(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－101054
（P2001－101054A）
(43)公開日　平成13年４月13日(2001.4.13)

(51)Int.Cl.'		識別記号	FI			テーマコード'(参考)
Ｇ０６Ｆ	12/00	５３７	Ｇ０６Ｆ	12/00	５３７Ａ	５Ｂ０１７
		５４５			５４５Ｂ	５Ｂ０４５
	12/14	３１０		12/14	３１０Ｋ	５Ｂ０８２
	15/00	３３０		15/00	３３０Ｂ	５Ｂ０８５
	15/16	６２０		15/16	６２０Ｓ	

審査請求　未請求　請求項の数17　ＯＬ　（全 35 頁）

(21)出願番号　特願平11－275702

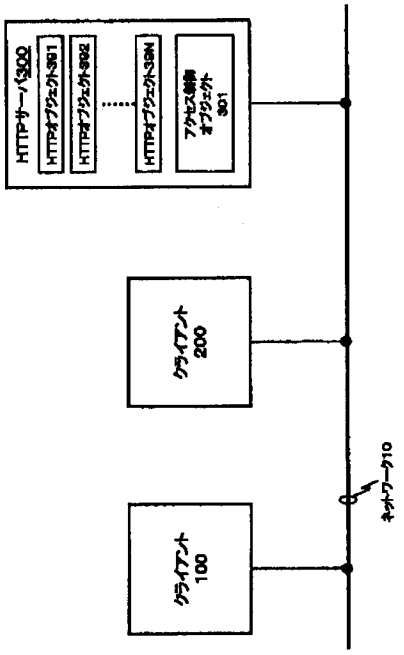
(22)出願日　平成11年９月29日(1999.9.29)

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号
(72)発明者　堀切　和典
神奈川県足柄上郡中井町境430　グリーンテクなかい　富士ゼロックス株式会社内
(74)代理人　100086531
弁理士　澤田　俊夫

Ｆターム(参考)　5B017 AA01 BA06 BB06 CA16
5B045 BB47 GG01 JJ33
5B082 EA11
5B085 AE00 AE01 AE04

(54)【発明の名称】　アクセス権限委譲方法

(57)【要約】
【課題】　サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関する権限情報をサブジェクト（ユーザ）間で安全に転送する。
【解決手段】　クライアントとサーバ間では、クライアントのユーザ情報と秘密情報が共有されている。権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成し、生成した権限情報と秘密情報とを連結したビット列に対して一方向性関数や暗号化関数を適用することで、秘密情報を知らない第３者が改竄不能な保護化権限情報を生成する。保護化権限情報を用いることで、アクセス権限を安全に委譲することができる。また、サーバは、秘密情報を用いて保護化権限情報を解析することで、オブジェクト要求するクライアントが正当か否かを安全に確認することができる。



1

【特許請求の範囲】

【請求項１】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（ｃ）クライアントが権限情報を生成するステップと、（ｄ）クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、（ｅ）クライアントが、ユーザ情報と権限情報と保護化権限情報を他のクライアントに送信するステップと、（ｆ）他のクライアントが、ユーザ情報と権限情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（ｇ）サーバが、ステップ（ｆ）において受信した権限情報が有効か否かを検査するステップと、（ｈ）サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、（ｉ）サーバが、ステップ（ｆ）において受信した保護化権限情報とステップ（ｈ）において生成した保護化権限情報とを比較するステップと、（ｊ）ステップ（ｉ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項２】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するクライアントが、他のクライアントに対してアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）サーバと共有すべきユーザ情報と秘密情報を保持するステップと、（ｂ）権限情報を生成するステップと、（ｃ）少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して、他のクライアントに安全に委譲することができる保護化権限情報を生成するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項３】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するサーバが、アクセス権限を委譲されたクライアントからのアクセス要求に対して安全に応答するためのアクセス権限委譲方法であって、（ｆ）ユーザ情報と権限情報と保護化権限情報を含んだアクセス要求を受信するステップと、（ｇ）ステップ（ｆ）において受信した権限情報が有効か否かを検査するステップと、（ｈ）少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、（ｉ）ステップ（ｆ）において受信した保護化権限情報とステップ（ｈ）において生成した保護化権限情報とを比較するステップと、（ｊ）ステップ（ｉ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスを許容するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項４】前記の所定の演算操作とは、被演算子の各々を連結したビット列に対して一方向性関数を適用することであることを特徴とする請求項１、２、又は３のいずれかに記載のアクセス権限委譲方法。

【請求項５】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（Ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（Ｃ）クライアントが権限情報を生成するステップと、（Ｄ）クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第１の保護化権限情報を生成するステップと、（Ｅ）クライアントが、ユーザ情報と権限情報と第１の保護化権限情報を他のクライアントに送信するステップと、（Ｆ）他のクライアントが、サーバからチャレンジ文字列を受信するステップと、（Ｇ）他のクライアントが、少なくともチャレンジ文字列と第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｈ）他のクライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（Ｈ）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、少なくともチャレンジ文字列とステップ（Ｊ）において生成した第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（Ｈ）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）

における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項６】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、ユーザ情報と権限情報と第１の保護化権限情報を譲受したクライアントとサーバとの間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ｆ）サーバが、オブジェクトへのアクセスを要求するクライアントに対してチャレンジ文字列を送信するステップと、（Ｇ）クライアントが、少なくともチャレンジ文字列と第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｈ）クライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（Ｈ）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、少なくともチャレンジ文字列とステップ（Ｊ）において生成した第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（Ｈ）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）における比較の結果、両者が一致することに応答して、サーバがオブジェクトに対するアクセスを許容するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項７】前記の所定の演算操作とは、被演算子の各々を連結したビット列に対して一方向性関数を適用することであることを特徴とする請求項　５又は６のいずれかに記載のアクセス権限委譲方法。

【請求項８】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（ｃ）クライアントが権限情報を生成するステップと、（ｄ）クライアントが、秘密情報を用いて権限情報を暗号化して保護化権限情報を生成するステップと、（ｅ）クライアントが、ユーザ情報と保護化権限情報を他のクライアントに送信するステップと、（ｆ）他のクライアントが、ユーザ情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（ｇ）サーバが、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成するステップと、（ｈ）サーバが、ステップ（ｇ）において生成した権限情報が有効か否かを検査するステップと、（ｉ）ステップ（ｈ）における有効性の検査結果に従って、オブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項９】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するクライアントが、他のクライアントに対してアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）サーバと共有すべきユーザ情報と秘密情報を保持するステップと、（ｂ）権限情報を生成するステップと、（ｄ）秘密情報を用いて権限情報を暗号化して、他のクライアントに安全に委譲することができる保護化権限情報を生成するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項１０】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するサーバが、アクセス権限を委譲されたクライアントからのアクセス要求に対して安全に応答するためのアクセス権限委譲方法であって、（ｆ）ユーザ情報と保護化権限情報を含んだアクセス要求を受信するステップと、（ｇ）ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成するステップと、（ｈ）ステップ（ｇ）において生成した権限情報が有効か否かを検査するステップと、（ｉ）ステップ（ｈ）における有効性の検査結果に従って、オブジェクトに対するアクセスを許容するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項１１】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（Ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（Ｃ）クライアントが権限情報

を生成するステップと、（Ｄ）クライアントが、秘密情報を用いて権限情報を暗号化して第１の保護化権限情報を生成するステップと、（Ｅ）クライアントが、ユーザ情報と権限情報と第１の保護化権限情報を他のクライアントに送信するステップと、（Ｆ）他のクライアントが、サーバからチャレンジ文字列を受信するステップと、（Ｇ）他のクライアントが、第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｈ）他のクライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（II）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、秘密情報を用いて権限情報を暗号化して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、ステップ（Ｊ）において生成した第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（Ｈ）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項１２】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、ユーザ情報と権限情報と第１の保護化権限情報を譲受したクライアントとサーバとの間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ｆ）サーバが、オブジェクトへのアクセスを要求するクライアントに対してチャレンジ文字列を送信するステップと、（Ｇ）クライアントが、第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｈ）クライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（Ｈ）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、秘密情報を用いて権限情報を暗号化して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、ステップ（Ｊ）において生成した第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（II）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）における比較の結果、両者が一致することに応答して、サーバがオブジェクトに対するアクセスを許容するステップと、を具備することを特徴とするアクセス権限委譲方法。

【請求項１３】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、
第１のクライアントが第２のクライアントに秘密情報を送信するステップと、
第１のクライアントが第２のクライアントに暗号化鍵を送信するステップと、
第２のクライアントが、前記暗号化鍵を用いて前記秘密情報を暗号化した後に、２次記憶装置に格納するステップと、を具備することを特徴とする情報管理方法。

【請求項１４】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、
第１のクライアントが、暗号化鍵を用いて秘密情報を暗号化して、保護化秘密情報を生成するステップと、
第１のクライアントが第２のクライアントに前記保護化秘密情報を送信するステップと、
第２のクライアントが前記保護化秘密情報を２次記憶装置に格納するステップと、
第１のクライアントが、前記暗号化鍵で暗号化された情報を復号化するための復号化鍵を、第２のクライアントに送信するステップと、
第２のクライアントが、復号化鍵を用いて保護化秘密情報を復号化して、秘密情報を得るステップと、を具備することを特徴とする情報管理方法。

【請求項１５】前記暗号化鍵と前記復号化鍵は同一鍵であることを特徴とする請求項１４に記載の情報管理方法。

【請求項１６】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、
第１のクライアントが第２のクライアントに秘密情報を送信するステップと、
第２のクライアントが、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを保持するステップと、
第２のクライアントが復号化鍵を第１のクライアントに送信するステップと、
第２のクライアントが、前記秘密情報を前記暗号化鍵で暗号化した保護化秘密情報を２次記憶装置に格納するステップと、
第２のクライアントが、前記復号化鍵を用いて保護化秘

密情報を復号化して前記秘密情報を得るステップと、を具備することを特徴とする情報管理方法。

【請求項１７】オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第１のクライアントが第２のクライアントに第１の秘密情報を送信するステップと、

第２のクライアントが第１のクライアントにチャレンジ文字列を送信するステップと、

第１のクライアントが前記チャレンジ文字列と第２の秘密情報に対して所定の演算処理を適用して暗号鍵を生成するステップと、

第１のクライアントが第２のクライアントに前記暗号鍵を送信するステップと、

第２のクライアントが、前記暗号鍵を用いて前記秘密情報を暗号化して得た保護化秘密情報を２次記憶装置に格納するステップと、を具備することを特徴とする情報管理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送するアクセス権限委譲方法に関する。

【０００２】更に詳しくは、本発明は、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をホスト（ユーザ）間で安全に転送するための、アクセス権限委譲方法に関する。

【０００３】

【従来の技術】昨今の情報技術（ＩＴ）分野における急速な進歩とともに、ワークステーションやパーソナル・コンピュータなどの各種の汎用コンピュータ・システムが開発・製造され、大学等の研究機関や企業のオフィス、一般家庭内に広く普及している。

【０００４】コンピュータ・システム上では、テキスト形式の文書ファイルや、音声ファイル、画像ファイルなど、デジタイズされた様々な資源オブジェクトを取り扱うことができる。

【０００５】最近では、多くのコンピュータ・システムは、ＬＡＮ（Ｌｏｃａｌ　ＡｒｃａＮｅｔｗｏｒｋ）やインターネットなどのネットワークに接続され、分散コンピューティング環境に置かれている。分散コンピューティング環境下では、各ユーザはプログラムやデータなど資源オブジェクトの所在を特に認識する必要がなくなる。さらに、コンピュータにおいて実行される手続きやメソッドも、ネットワーク上で分散して保持され、管理されている。

【０００６】例えば、ネットワーク上のある１つのコンピュータ上で動作しているプロセスが、他のコンピュータ上で動作するプロセスの手続きを呼び出して実行させるという、「遠隔手続き呼び出し（ＲＰＣ：Ｒｅｍｏｔｅ　Ｐｒｏｃｅｄｕｒｅ　Ｃａｌｌ）」、若しくは、「遠隔メソッド呼び出し（ＲＭＩ：Ｒｅｍｏｔｅ　Ｍｅｔｈｏｄ　Ｉｎｖｏｃａｔｉｏｎ）」と呼ばれる手法も広汎に採り入れられている。実行される手続きのインターフェースを予め記述し、呼び出し側及び実行側の双方のコンピュータに配置しておくことによって、このような遠隔手続き呼び出しを好適に実現することができる。

【０００７】その反面、ファイルなどの夫々の共有資源オブジェクト（Ｏｂｊｅｃｔ）に対して、誰（Ｓｕｂｊｅｃｔ）にどの程度のアクセス操作（Ｖｅｒｂ）を許容するか、といった「アクセス権（Ａｃｃｅｓｓ　Ｐｒｉｖｉｌｅｇｅｓ）」に関する管理方法や管理システムが重要な技術的課題となってくる。アクセス権の制御は、Ｓ（Ｓｕｂｊｅｃｔ：主体）－Ｖ（Ｖｅｒｂ：アクセス操作）－Ｏ（Ｏｂｊｅｃｔ：操作対象としてのファイルなど）の関係に見立てて記述し、管理することができる。Ｓすなわちサブジェクトとは、ファイル操作を望むユーザ、ユーザ・アカウント、又はユーザ識別情報とほぼ同義と把握されたい。

【０００８】ここで言うオブジェクト（Ｏ）に対するアクセス操作（Ｖ）の権限としては、例えば以下のようなものが挙げられる。

（１）ファイルを読み出す権利

（２）ファイルに書き込む（更新する）権利

（３）ファイルを新規作成する権利

（４）ファイルを削除する権利

（５）ファイルを検索する権利

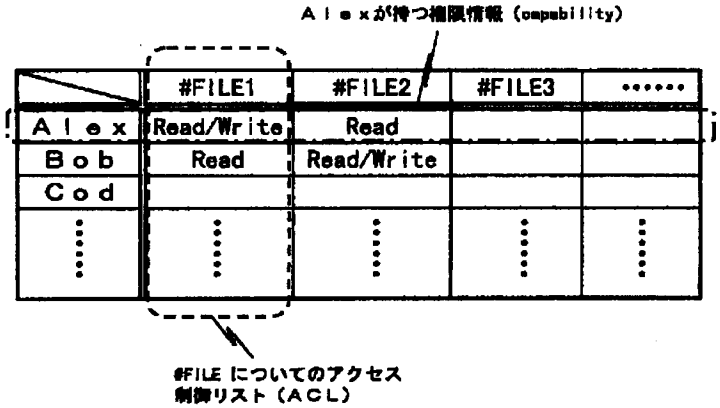
（６）ファイル名などのファイル属性を変更する権利

【０００９】全てのユーザに対して無制限なアクセス権を許容してしまうと、ファイルが勝手に複製、改竄、又は削除されてしまうなど悪用や乱用が横行する危険が発生する。この結果、あるユーザは予期しない不利益を被り、時として致命傷を負う。アクセス権を適切に管理しなければ、分散コンピューティング環境の崩壊をも招来するであろう。

【００１０】オブジェクトに対するアクセス権の管理方法として、「アクセス制御マトリックス」なる概念が、当業界において古くから知られている。このアクセス制御マトリックスは、以下の［表１］に示すように、誰（Ｓｕｂｊｅｃｔ）に、ファイルなどの各オブジェクト（Ｏｂｊｅｃｔ）に対してどの程度のアクセス権（Ｖｅｒｂ）を与えたかを一覧表示したテーブル形式で記述される。

【００１１】

【表１】



【００１２】［表１］に示す例では、例えば”Ａｌｅｘ”なる主体すなわちユーザ・アカウントには、オブジェクトとしての”Ｆｉｌｅ＃１”に対する読み出し（Ｒｅａｄ）及び書き込み（Ｗｒｉｔｅ）というアクセス操作が許容されているが、”Ｆｉｌｅ＃２”に対しては読み出しのみ許され…、”Ｂｏｂ”なる主体には…となっている。

【００１３】アクセス制御マトリックスを各カラム毎に切り出したものは、オブジェクトが各サブジェクトすなわちユーザに対して許容した操作権限を記述した情報であり、「アクセス制御リスト（Ａｃｃｅｓｓ Ｃｏｎｔｒｏｌ Ｌｉｓｔ：ＡＣＬ）」と呼ばれる。例えば、［表１］に示した例では、”Ｆｉｌｅ＃１”に相当するカラムは、該オブジェクトに対してＡｌｅｘ，Ｂｏｂ，Ｃｏｄの各々に与えられた操作権限を示したアクセス制御リストである。サーバ用若しくは開発プラットフォーム用のオペレーティング・システム（ＯＳ）として広汎に普及している”Ｕｎｉｘ”は、簡略形式のアクセス制御リストを使用している。

【００１４】また、アクセス制御マトリックスを各行毎に切り出したものは、サブジェクトすなわちユーザが各々のオブジェクトに対して許容された操作権限を記述した情報であり、「権限情報」若しくは「ケイパビリティ（Ｃａｐａｂｉｌｉｔｙ）」と呼ばれる。例えば、「表１」に示すアクセス制御マトリックスの第１行はユーザ”Ａｌｅｘ”に関するケイパビリティに相当し、また、第２行はユーザ”Ｂｏｂ”に関するケイパビリティに相当する。

【００１５】ケイパビリティは、オブジェクトへのアクセスを許容する一種の鍵の役割を持つ。すなわち、ケイパビリティを所有するサブジェクトすなわちユーザには、オブジェクトへのアクセスが許可される。例えば、無数のホストがＴＣＰ／ＩＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ Ｐｒｏｔｏｃｏｌ）接続されてなるネットワーク（例えばインターネット）上では、ケイパビリティをＵＲＬ（Ｕｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒ）文字列の中で記述して、ＨＴＴＰ（Ｈｙｐｅｒ Ｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ）メッセージとしてホスト間で交換することもできる。

【００１６】例えば、ＡｌｅｘがＣｏｄにケイパビリティを渡すことによって、Ｃｏｄは、各オブジェクトに対してＡｌｅｘと同等の操作権限を所有することができる。ケイパビリティを引き継いだＣｏｄは、Ａｌｅｘの不在中も、Ａｌｅｘの代理として各オブジェクトにアクセスして、Ａｌｅｘの任務を代行することができる。

【００１７】しかしながら、ケイパビリティを無制限に譲渡しあるいは貸し与えてしまうと、ケイパビリティの悪用・乱用により、思わぬ不利益を受ける可能性がある。このため、ケイパビリティの譲渡者は、ケイパビリティに有効期限や使用許可回数を付加したり、オブジェクトの操作権限を制限する（例えばフル・アクセスを読み出しのみに限定する）など、権限内容を弱めてケイパビリティを渡すことが好ましい。

【００１８】また、ケイパビリティを譲り受けたサブジェクトが、ケイパビリティの内容を簡単に解釈できるような形式で渡したのでは、譲受者はケイパビリティの内容を改竄して権限を勝手に強めたり、無断でケイパビリティを複製して、オブジェクトを無制限にアクセス操作してしまうであろう。また、ケイパビリティの転送を傍受する第三者も、同様に、オブジェクトを不正に操作できてしまう。

【００１９】以上を要言するならば、ネットワーク上の各ホスト間でケイパビリティを安全に転送することがオブジェクトを保護する上で重要な技術的課題となる訳である。

【００２０】オブジェクトのアクセス制御に関しては、既に幾つかの技術が提案されている。

【００２１】例えば特開平５－８１２０４号公報には、分散型コンピュータ・システムにおけるアクセス制御について開示されている。同公報によれば、ケイパビリティに相当する特権属性証明書（ＰＡＣ）の代理使用を制御すると同時に、ＰＡＣを多くの目的のために使用できる方法が提供されている。すなわち、ＰＡＣを配布する

際には、これに開始者実体に相当する開始者資格付与属性を含めるようにするとともに、暗号的に開始者資格付与属性を有した暗号キーを開始者実体に配布するようになっている。

【００２２】しかしながら、特開平５－８１２０４号公報に開示された方法によれば、使用期限やアクセス回数など、ケイパビリティのバリエーションには対応できない。また、開始実体者がケイパビリティを弱めたＰＡＣを自由に作成する方法については一切言及していない。

【００２３】また、特開平９－３１９６５９号公報には、非分散型のコンピュータ・システムにおいて、各ユーザに異なるケイパビリティを割り当てる方式について開示されている。同公報によれば、コンピュータ機能の全体はケイパビリティ・セットを有するイベント・セットに細分化されており、ユーザがコンピュータ・システム上で実行しようとする特定のジョブに応じてケイパビリティが与えられるようになっている。

【００２４】しかしながら、特開平９－３１９６５９号公報に係る発明は、ケイパビリティを安全に守ることができない分散環境に適用されたものではない。また、同公報は、オブジェクト間でケイパビリティの検査を安全に行う方法については一切言及していない。

【００２５】また、特開平９－２５１４２５号公報には、分散システムにおけるシステム資源へのアクセスのセキュリティ制御について開示している。同公報に開示されるセキュリティ制御システムによれば、グループ識別標識を記憶しておき、これをターゲット・オブジェクトに結び付け、次にメンバーシップ検査を使用して、ターゲット・オブジェクトへのアクセスを要求するクライアントがターゲットに対するアクセス権限を有するグループ・メンバーであるか否かを判断するようになっている。

【００２６】しかしながら、特開平９－２５１４２５号公報では、使用期限やアクセス回数など、ケイパビリティのバリエーションには対応する手法については開示していない。また、あるケイパビリティの保有者が新たにケイパビリティを生成したりこれを無効化する手法についても一切言及していない。

【００２７】また、分散ＯＳ（オペレーティング・システム）として当業界では周知の”Ａｍｏｅｂａ”は、譲渡可能で且つ偽造が困難なケイパビリティを提供している。これは以下の機構により実現される。

（１）クライアントがオブジェクト生成要求をサーバに送信する。

（２）サーバは、オブジェクトを生成して、オブジェクト番号と乱数を割り当てる。乱数は、オブジェクト番号でインデックス付けされたオブジェクト・テーブルに格納される。

（３）サーバは、乱数をキーとして権限と乱数を暗号化したチェック・フィールドを含むケイパビリティを生成する。

（４）ケイパビリティをクライアントに返す。

（５）クライアントが、サーバに対して、ケイパビリティを提示したアクセス要求を出す。

（６）サーバは、ケイパビリティからオブジェクト番号とチェック・フィールドを抽出し、オブジェクト・テーブルに格納されている乱数を基にチェック・フィールドから権限と乱数を復号化する。

（７）復号の結果得られた乱数がオブジェクト・テーブルに格納された乱数と一致するかどうか検証する。

（８）要求された操作がケイパビリティに記述された権限に合致すれば、サーバはその操作を実行する。

【００２８】Ａｍｏｅｂａでは、さらに、ケイパビリティの保持者が権限を弱めた新しいケイパビリティを生成するために、以下の機構も備えている。

（１）クライアントとサーバがＮ個の可換な一方向性関数を共有する。

（２）クライアントが第２の権限を生成する。

（３）　クライアントが第１の権限と第２の権限との差分の権限に該当する番号と対応付けた一方向性関数を全て適用して、第２のチェック・フィールドを生成する。

（４）サーバが、第２の権限と第２のチェック・フィールドを含むケイパビリティを受信する。

（５）サーバは、権限フィールドに従い、一方向性関数を順次適用して、クライアントが提示したチェック・フィールドと一致した場合に、その操作を実行する。

【００２９】しかしながら、Ａｍｏｅｂａは、Ｎ個の可換な一方向性関数を用いてＮ個の権限の可否を記述する方法については規定しているが、使用権限や回数といった権限についての多くのバリエーションに対応する点については考慮していない。また、ケイパビリティの無効化に関しては、サーバで保持されている乱数を変更することによってあるオブジェクトについての全てのケイパビリティを無効化する手法が規定されているが、特定のケイパビリティを無効化する手法については言及していない。例えば、あるケイパビリティの保持者が生成したケイパビリティや、これから派生したケイパビリティのみを無効化する点は一切規定していない。また、Ａｍｏｅｂａではオブジェクトにアクセスするためのオブジェクト番号自体については一切プロテクトがかけられていない。

【００３０】また、Ｗｅｂ上で公開されたＢｊｏｒｎ Ｎ．　Ｆｒｅｅｍａｎ－Ｂｅｎｓｏｎの論文”Ｕｓｉｎｇ ｔｈｅ Ｗｅｂ ｔｏ Ｐｒｉｖａｔｅ Ｉｎｆｏｒｍａｔｉｏｎ －ｏｒ－ Ａ Ｓｈｏｒｔ Ｐａｐｅｒ Ａｂｏｕｔ Ｐａｓｓｗｏｒｄ Ｐｒｏｔｅｃｔｉｏｎ Ｗｉｔｈｏｕｔ Ｃｌｉｅｎｔ Ｍｏｄｉｆｉｃａｔｉｏｎ”（ＵＲＬは”ｈｔｔｐ：／／ｗｗｗ１．ｃｅｒｎ．ｃｈ／ｗｗｗ９４／ＰｒｅｌｉｍＰｒｏｃｓ．ｈｔｍｌ”）では、機密性のあるＵＲＬ（すなわちケイ

パビリティ）の取り扱いについて開示されている。同論文に記載された方式は以下の手順に従う。

（１）サーバがログイン名とパスワードの組からなるパスワード・リストを生成・保持する。また、サーバは暗号鍵を保持している。

（２）クライアントがログイン名とパスワードの組からなるメッセージをサーバに送信する。

（３）サーバは、パスワード・リストを検索し、クライアントから送信されたログイン名とパスワードの組が見つかれば、このログイン名とパスワードを暗号鍵で暗号化した文字列をアクセス・キーとして、ＵＲＬに含めてクライアントに送信する。

（４）クライアントは、受け取ったＵＲＬを用いてサーバにアクセス要求する。

（５）サーバは、受信したＵＲＬからアクセス・キーを抽出し、暗号鍵を用いて復号化して、もとのログイン名とパスワードの組を得る。そして、これがパスワード・リストに登録されているかどうかを検査する。

（６）登録されていれば、サーバは対応するオブジェクトへのアクセスを許可する。

【００３１】また、同論文では、クライアントがパスワードの変更を行うことでＵＲＬを無効化することができる点を言及している。

【００３２】しかしながら、同論文では、ある１つのオブジェクトについての正当で且つ異なるＵＲＬ（すなわちケイパビリティ）を複数生成する点については開示していない。

【００３３】以上を総括すると、従来技術では下記のような問題があった。すなわち、

（１）使用期限や有効使用回数などの権限内容に多種多様なバリエーションに対応しながらケイパビリティを安全に管理し委譲することが困難である。

（２）該当するオブジェクトに関する全てのケイパビリティを無効化するのではなく、一部のケイパビリティのみを無効化することは困難である。

（３）ケイパビリティを所有するサブジェクトが、ケイパビリティが持つ権限の内容を変更した（弱めた）ケイパビリティを自由に生成し、且つ、これを安全に管理し委譲し、あるいは無効化することができない。

【００３４】

【発明が解決しようとする課題】本発明の目的は、サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することにある。

【００３５】本発明の更なる目的は、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」を、ホスト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することにある。

【００３６】本発明の更なる目的は、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、他のサブジェクトに安全に委譲することができる、優れたアクセス権限委譲方法を提供することにある。

【００３７】本発明の更なる目的は、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、生成されたケイパビリティをオブジェクトの管理者が安全に検査することができる、優れたアクセス権限委譲方法を提供することにある。

【００３８】

【課題を解決するための手段及び作用】本発明は、上記課題を参酌してなされたものであり、その第１の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるコンピューティング環境下で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（ｃ）クライアントが権限情報を生成するステップと、（ｄ）クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、（ｅ）クライアントが、ユーザ情報と権限情報と保護化権限情報を他のクライアントに送信するステップと、（ｆ）他のクライアントが、ユーザ情報と権限情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（ｇ）サーバが、ステップ（ｆ）において受信した権限情報が有効か否かを検査するステップと、（ｈ）サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、（ｉ）サーバが、ステップ（ｆ）において受信した保護化権限情報とステップ（ｈ）において生成した保護化権限情報とを比較するステップと、（ｊ）ステップ（ｉ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法である。

【００３９】また、本発明の第２の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上

で、各クライアント間及びクライアント－サーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（Ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（Ｃ）クライアントが権限情報を生成するステップと、（Ｄ）クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第１の保護化権限情報を生成するステップと、（Ｅ）クライアントが、ユーザ情報と権限情報と第１の保護化権限情報を他のクライアントに送信するステップと、（Ｆ）他のクライアントが、サーバからチャレンジ文字列を受信するステップと、（Ｇ）他のクライアントが、少なくともチャレンジ文字列と第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｈ）他のクライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（Ｈ）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、少なくともチャレンジ文字列とステップ（Ｊ）において生成した第１の保護化権限情報とからなる情報に対して所定の演算操作を適用して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（Ｈ）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法である。

【００４０】また、本発明の第３の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアント－サーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（ｃ）クライアントが権限情報を生成するステップと、（ｄ）クライアントが、秘密情報を用いて権限情報を暗号化して保護化権限情報を生成するステップと、（ｅ）クライアントが、ユーザ情報と保護化権限情報を他のクライアントに送信するステップと、（ｆ）他のクライアントが、ユーザ情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（ｇ）サーバが、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成するステップと、（ｈ）サーバが、ステップ（ｇ）において生成した権限情報が有効か否かを検査するステップと、（ｉ）ステップ（ｈ）における有効性の検査結果に従って、オブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法である。

【００４１】また、本発明の第４の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアント－サーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、（Ａ）各クライアントがユーザ情報と秘密情報を保持するステップと、（Ｂ）サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、（Ｃ）クライアントが権限情報を生成するステップと、（Ｄ）クライアントが、秘密情報を用いて権限情報を暗号化して第１の保護化権限情報を生成するステップと、（Ｅ）クライアントが、ユーザ情報と権限情報と第１の保護化権限情報を他のクライアントに送信するステップと、（Ｆ）他のクライアントが、サーバからチャレンジ文字列を受信するステップと、（Ｇ）他のクライアントが、第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｈ）他のクライアントが、ユーザ情報と権限情報と第２の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、（Ｉ）サーバが、ステップ（Ｈ）において受信した権限情報が有効か否かを検査するステップと、（Ｊ）サーバが、秘密情報を用いて権限情報を暗号化して第１の保護化権限情報を生成するステップと、（Ｋ）サーバが、ステップ（Ｊ）において生成した第１の保護化権限情報を用いてチャレンジ文字列を暗号化して第２の保護化権限情報を生成するステップと、（Ｌ）サーバが、ステップ（Ｈ）において受信した第２の保護化権限情報とステップ（Ｋ）において生成した第２の保護化権限情報とを比較するステップと、（Ｍ）ステップ（Ｎ）における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、を具備することを特徴とするアクセス権限委譲方法である。

【００４２】また、本発明の第５の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアント－サーバ間で秘密情報を安全に管理するための情報管理方法であって、第１のクライアントが第２のクライアントに秘密情報を送信するステップと、第１のク

ライアントが第２のクライアントに暗号化鍵を送信するステップと、第２のクライアントが、前記暗号化鍵を用いて前記秘密情報を暗号化した後に、２次記憶装置に格納するステップと、を具備することを特徴とする情報管理方法である。

【００４３】また、本発明の第６の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、第１のクライアントが、暗号化鍵を用いて秘密情報を暗号化して、保護化秘密情報を生成するステップと、第１のクライアントが第２のクライアントに前記保護化秘密情報を送信するステップと、第２のクライアントが前記保護化秘密情報を２次記憶装置に格納するステップと、第１のクライアントが、前記暗号化鍵で暗号化された情報を復号化するための復号化鍵を、第２のクライアントに送信するステップと、第２のクライアントが、復号化鍵を用いて保護化秘密情報を復号化して、秘密情報を得るステップと、を具備することを特徴とする情報管理方法である。

【００４４】本発明の第６の側面に係る情報管理方法において、前記暗号化鍵と前記復号化鍵は、同一鍵すなわち「対称鍵暗号方式」の鍵であっても、あるいは、暗号化鍵（秘密鍵）で暗号化した情報は復号化鍵（公開鍵）でしか復号化できないような「公開鍵暗号方式」における鍵の組み合わせであってもよい。

【００４５】また、本発明の第７の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、第１のクライアントが第２のクライアントに秘密情報を送信するステップと、第２のクライアントが、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを保持するステップと、第２のクライアントが復号化鍵を第１のクライアントに送信するステップと、第２のクライアントが、前記秘密情報を前記暗号化鍵で暗号化した保護化秘密情報を２次記憶装置に格納するステップと、第２のクライアントが、前記復号化鍵を用いて保護化秘密情報を復号化して前記秘密情報を得るステップと、を具備することを特徴とする情報管理方法である。

【００４６】また、本発明の第８の側面は、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、第１のクライアントが第２のクライアントに第１の秘密情報を送信するステップと、第２のクライアントが第１のクライアントにチャレンジ文字列を送信するステップと、第１のクライアントが前記チャレンジ文字列と第２の秘密情報に対して所定の演算処理を適用して暗号鍵を生成するステップと、第１のクライアントが第２のクライアントに前記暗号鍵を送信するステップと、第２のクライアントが、前記暗号鍵を用いて前記秘密情報を暗号化して得た保護化秘密情報を２次記憶装置に格納するステップと、を具備することを特徴とする情報管理方法である。

【００４７】

【作用】本発明は、例えばＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）やインターネットのような分散コンピューティング環境において適用される。かかる分散コンピューティング環境下では、オブジェクトを提供する１以上のサーバと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、いわゆる「オブジェクト空間」が形成されている。オブジェクト空間の一例は、各ホストがＴＣＰ／ＩＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ　Ｃｏｎｔｒｏｌ　Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ）接続された分散コンピューティング環境下で展開される、ＨＴＭＬ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ）形式で記述されたＨＴＴＰ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ）オブジェクトが提供されるＷＷＷ（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ）情報空間である。

【００４８】オブジェクト空間上では、各クライアントのオブジェクトに対するアクセス権限は「権限情報」若しくは「ケイパビリティ（ｃａｐａｂｉｌｉｔｙ）」という形式で記述される。したがって、クライアントは、基本的には、自身のユーザ情報と自身に付与された権限情報を、オブジェクトを管理し提供するサーバに提示することによって、オブジェクトへのアクセスが許容される。

【００４９】さらに、クライアントから権限情報を引き継いだ他のクライアントも、元のクライアントのユーザ情報と権限情報とを提示することによって、オブジェクトへのアクセスが許容される。しかしながら、無制限に権限情報を譲渡してしまうと、譲受者による権限情報の乱用や不充分な安全管理のために、譲渡者であるクライアントは、権限情報の悪用や乱用、あるいは権限情報の無断の複製や改竄によって、不測の不利益を被りかねない。

【００５０】そこで、権限情報の譲渡者は、オブジェクトに対する無制限のアクセスを排除するために、権限内容を弱めた権限情報（例えば、有効期限や使用許可回数などを付加したり、オブジェクトの操作権限を制限した権限情報）を委譲することが好ましい。また、委譲した後に、権限情報が勝手に改竄されないように（例えば、

勝手に有効期限を書き換えたり、操作権限を強化したりされないように）、権限情報を安全に送信できるように対策を講じる必要がある。

【００５１】本発明は、各クライアントとサーバ間で、クライアントのユーザ情報と秘密情報が共有されていることを前提とする。秘密情報は、パスワード等を指し、例えば所定の情報に暗号化関数を適用する際の秘密鍵として利用することができる。

【００５２】本発明の第１の側面によれば、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成し、生成した権限情報と秘密情報とを連結したビット列に対して所定の演算操作を適用することで、改竄などの悪用の危険を排した保護化権限情報を生成することができる。

【００５３】ここで言う所定の演算操作の一例は「一方向性関数」である。一方向性関数は、逆関数を求めることが極めて困難な関数であり、該関数を適用する前の引数の値を推測することを不可能にする作用がある。したがって、秘密情報を知らない第３者が保護化権限情報を勝手に改竄することはできない。

【００５４】クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、保護化権限情報とを送信する。そして、他のクライアントは、譲受したユーザ情報、権限情報、及び保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を行うことができる。

【００５５】一方のサーバは、まず、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

【００５６】次いで、サーバは、アクセス要求メッセージで受け取ったユーザ情報と秘密情報とを連結したビット列に対して、所定の演算操作、例えば一方向性関数を適用して、保護化権限情報を再現してみる。そして、この再現された保護化権限情報を、アクセス要求メッセージで受け取った保護化権限情報と比較照合することで、アクセス要求クライアントが正当な権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

【００５７】また、本発明の第２の側面に係るアクセス権限委譲方法では、認証処理のために、「チャレンジ文字列」と呼ぶ１回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を採用する。

【００５８】まず、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成する。そして、生成した権限情報と秘密情報とを連結したビット列に対して一方向性関数などの演算操作を適用することで、悪用の危険を排した第１の保護化権限情報を生成する。秘密情報を知らない第３者は、第１の保護化権限情報を勝手に改竄することはできない。

【００５９】クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、第１の保護化権限情報を送信する。

【００６０】サーバは、クライアントからのアクセス要求に対して、チャレンジ文字列を返信する。これに対し、クライアントは、チャレンジ文字列と第１の保護化権限情報を連結したビット列に対して一方向性関数などの所定の演算操作を適用して、悪用の危険を排した第２の保護化権限情報を生成する。チャレンジ文字列を知らない第３者は、第２の保護化権限情報を勝手に改竄することはできない。

【００６１】次いで、クライアントは、譲受したユーザ情報、権限情報、及び第２の保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を改めて行う。

【００６２】サーバは、アクセス要求に応答して、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

【００６３】次いで、サーバは、アクセス要求メッセージで受け取ったユーザ情報と秘密情報とを連結したビット列に対して一方向性関数などの所定の演算操作を適用して、第１の保護化権限情報を再現してみる。さらに、チャレンジ文字列と第１の保護化権限情報を連結したビット列に対して一方向性関数などの所定の演算操作を適用して、第２の保護化権限情報を再現してみる。そして、この再現された第２の保護化権限情報を、アクセス要求メッセージで受け取った第２の保護化権限情報と比較照合することで、アクセス要求クライアントが正当な権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

【００６４】また、本発明の第３の側面に係るアクセス権限委譲方法は、委譲する権限情報を保護化する所定の演算操作として、一方向性関数ではなく暗号化関数を用いたものである。クライアントとサーバの間で共有される秘密情報を暗号化関数の暗号鍵として用いることができる。暗号化方式としては、対称秘密鍵方式又は公開鍵暗号方式[*/]　の双方を適用することができる。

【００６５】まず、権限情報を委譲するクライアント

は、自身が持つ権限内容を弱めた権限情報を生成し、次いで、秘密情報を用いて権限情報を暗号化して、悪用の危険を排した保護化権限情報を生成する。したがって、秘密情報を知らない第３者は、保護化権限情報を勝手に改竄することはできない。

【００６６】クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と保護化権限情報を送信する。そして、他のクライアントは、譲受したユーザ情報と保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を行う。

【００６７】一方のサーバは、まず、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成する。

【００６８】次いで、サーバは、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

【００６９】また、本発明の第４の側面に係るアクセス権限委譲方法は、第３の側面に対していわゆる「チャレンジ文字列」と呼ぶ１回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を適用したものである。「チャレンジ・レスポンス認証」は「ゼロ知識証明」の一種である（前述）。

【００７０】まず、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成する。そして、秘密情報を用いて権限情報を暗号化して、悪用の危険を排した第１の保護化権限情報を生成する。秘密情報を知らない第３者は、第１の保護化権限情報を勝手に改竄することはできない。

【００７１】クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、第１の保護化権限情報を送信する。

【００７２】サーバは、クライアントからのアクセス要求に対して、チャレンジ文字列を返信する。これに対し、クライアントは、第１の保護化権限情報を用いてチャレンジ文字列を暗号化して、悪用の危険を排した第２の保護化権限情報を生成する。チャレンジ文字列を知らない第３者は、第２の保護化権限情報を勝手に改竄することはできない。

【００７３】次いで、クライアントは、譲受したユーザ情報、権限情報、及び第２の保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を改めて行う。

【００７４】サーバは、アクセス要求に応答して、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

【００７５】次いで、サーバは、秘密情報を用いて権限情報を暗号化して、第１の保護化権限情報を再現してみる。さらに、再現された第１の保護化権限情報を用いてチャレンジ文字列を暗号化して、第２の保護化権限情報を再現してみる。そして、この再現された第２の保護化権限情報を、アクセス要求メッセージで受け取った第２の保護化権限情報と比較照合することで、アクセス要求クライアントが正当な権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

【００７６】上述したように、本発明の第１乃至第４の側面に係るアクセス権限委譲方法によれば、クライアントは、悪用の危険を排した保護化権限情報を送信することで他のクライアントに対してアクセス権限を安全に委譲することができる。しかしながら、アクセス権限を譲受した他のクライアントが、保護化権限情報をハード・ディスクのような２次記憶装置に無防備に格納してしまうと、第三者がハード・ディスクを攻撃したり不正侵入したりして、保護化権限情報自体が漏洩してしまう危険がある。漏洩した結果、第三者による保護化権限情報を用いた不正アクセスを回避することは困難となってしまう。

【００７７】本発明の第５乃至第８の側面に係る情報管理方法は、このような技術的課題を鑑みたものであり、保護化権限情報のような秘密性の高い情報を２次記憶装置に安全に格納する方法を提供するものである。

【００７８】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００７９】《注釈》

＊：対称秘密鍵方式とは、通信の相手どうしで同一の秘密鍵を共有する方式であり、暗号化するために用いた秘密鍵と同じ鍵を用いて暗号化情報を復号化することができる。これに対し、公開鍵暗号方式とは、一方の鍵で暗号化した暗号化情報は他方の鍵でしか復号化できないという性質を持つ２個の鍵の組み合わせて暗号化する方式であり、一方の鍵をユーザ個人が秘密裏に保持する「秘密鍵」として保持し、他方の鍵を第３者に公開する「公開鍵」として用いるのが一般的である。例えば公開鍵を用いて暗号化することによって秘密鍵の所有者に秘密文書を安全に送信することができる。また、秘密鍵を用いて暗号化した署名を送信することにより、受信者は公開鍵を用いて署名を認証することができる。

【００８０】

【発明の実施の形態】本発明は、例えばＬＡＮ（Ｌｏｃ

ａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）やインタネットのような分散コンピューティング環境において適用される。ネットワーク上の各コンピュータ・システムすなわち「ホスト」は、ＴＣＰ／ＩＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ　Ｃｏｎｔｒｏｌ　Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ）のような所定の通信プロトコルに従って相互接続されている。

【００８１】かかる分散コンピューティング環境下では、オブジェクトを提供する１以上のサービス・オブジェクトと、オブジェクトを要求する１以上のクライアントとがネットワーク接続され、いわゆる「オブジェクト空間」が形成されている。例えば、インターネット上で公開されている広域情報提供システムWWW（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ）では、HTML（Ｈｙｐｅｒ　Ｔｅｘｔ　ＭａｒｋｕｐＬａｎｇｕａｇｅ）形式で記述されたハイパーテキスト・オブジェクトが、ＨＴＴＰ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ）プロトコルに従って提供される。

【００８２】各オブジェクトは、ＵＲＬ（Ｕｎｉｆｏｒｍ　Ｒｅｓｏｕｒｃｅ　Ｌｏｃａｔｏｒ）と呼ばれる所定書式の文字列によってオブジェクト空間上で一意に指定される。ＵＲＬとは、ネットワークなどの資源空間上で資源の名称やその在り処を特定するための文字列であり、スキーム名（プロトコル名）：／／ホスト名（ドメイン名）：ポート番号／パス名（ファイル名）」という形式で記述される（周知）。クライアントは、所望のＨＴＴＰオブジェクトに関するＵＲＬを含んだＨＴＴＰメッセージを送信することによって、ＨＴＴＰサーバに対してオブジェクトへのアクセス（すなわちオブジェクトに対する操作）を要求することができる。

【００８３】このようなオブジェクト空間上では、各クライアントのＨＴＴＰオブジェクトに対するアクセス権限は「権限情報」若しくは「ケイパビリティ（ｃａｐａｂｉｌｉｔｙ）」という形式で記述される。各クライアントは、基本的には、自身のユーザ情報と自身に付与された権限情報をサーバすなわちサービス・オブジェクトに提示することによって、オブジェクトへのアクセスが許容される。ＨＴＴＰプロトコルでは、ケイパビリティをＵＲＬ文字列の中で記述して、ＨＴＴＰメッセージとしてホスト間で交換することも許容される。

【００８４】また、クライアントは、権限情報を含んだＵＲＬを他のクライアントに引き渡すという形式で、オブジェクトに対するアクセス権限を委譲することができる。但し、既に［従来の技術］の欄でも説明したように、無制限に権限情報を譲渡してしまうと、譲渡者であるクライアントは、権限情報の悪用や乱用、権限情報の無断の複製や改竄によって、不測の不利益を被りかねない。

【００８５】以下で説明するように、本発明の各実施形態では、権限情報の譲渡者は、オブジェクトに対する不正なアクセスを排除するために、権限内容を弱めた権限情報（例えば、有効期限や使用許可回数などを付加したり、オブジェクトの操作権限を弱めた権限情報）を委譲することにした。また、委譲した後に、権限情報が勝手に改竄されないように（例えば、勝手に有効期限を書き換えたり、操作権限を強化したりされないように）、権限情報を安全に送信できるように対策を講じた。

【００８６】１．第１の実施形態

図１には、本発明の第１の実施形態に係る分散コンピューティング環境を模式的に示している。第１の実施形態は、権限情報に一方向性関数ＭＤを適用することによってアクセス権限の安全な委譲を実現するものである。以下、各部について説明する。

【００８７】インターネットのようなネットワーク１０上には、無数のホストがＴＣＰ／ＩＰ接続されている。ホストの一部は、ＨＴＴＰオブジェクトを提供するＨＴＴＰサーバ３００…であり、また、他の一部は、サーバに対してオブジェクトへのアクセスを要求するクライアント１００，２００…である。なお、サーバの実体は、ＨＴＴＰサーバ・アプリケーションを稼動する汎用コンピュータであり、また、クライアントは、WWWブラウザなどのクライアント・アプリケーションを稼動する汎用コンピュータである。

【００８８】図１に示す例では、ＨＴＴＰサーバ３００は、”ｈｔｔｐ：／／ｗｗｗ３００”なるＵＲＬで表されるものとする。また、ＨＴＴＰサーバ３００は、アクセス制御オブジェクト３０１と、参照番号３９１～３９Ｎで示されるＮ個のＨＴＴＰオブジェクトを所有している。各ＨＴＴＰオブジェクト３９１～３９Ｎはそれぞれ”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９１”，…，”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９Ｎ”なるＵＲＬで示されるものとする。また、アクセス制御オブジェクト３０１は、各ＨＴＴＰオブジェクト３９１～３９Ｎに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１～３９Ｎと同じサーバ３００上に存在する必要は必ずしもなく、例えば別のホスト（図示しない）上に存在してＨＴＴＰサーバ３００から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【００８９】各クライアント１００…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。図１に示す例では、クライアント１００は、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。

【００９０】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…に関するユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」（仮称）として保

管している。

【００９１】本実施形態では、ＨＴＴＰ１．１のＢａｓｉｃ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ（基本認証法）に従ってＨＴＴＰメッセージが交換されるものとする。すなわち、クライアント１００は、以下に示すようなＨＴＴＰリクエストを発行することにより、ＨＴＴＰオブジェクト３９１のＧＥＴメソッドを起動することができる。

【００９２】

【数１】

ＧＥＴ　／ｏｂｊｃｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｂａｓｉｃ　ａｕｔｈ　ｄａｔａ

【００９３】但し、”ａｕｔｈｄａｔａ”は、ユーザ情報とパスワードを連結したビット列ｕｓｅｒｉｄ１：ｐａｓｓｗｏｒｄ１を、改行のないｂａｓｅ６４エンコード法でエンコードした文字列である。なお、ここで言う”ｂａｓｅ６４”とは、ＭＩＭＥ（Ｍｕｌｔｉｐｕｒｐｏｓｅ　Ｉｎｔｅｒｎｅｔ　Ｍａｉｌ　Ｅｘｔｅｎｓｉｏｎｓ）を利用した電子メールの添付ファイルの符号化方式であり、文字列や添付ファイルを全て６４個のアルファベットや数字などのＡＳＣＩＩ（Ａｍｅｒｉｃａｎ　Ｓｔａｎｄａｒｄ　Ｃｏｄｅ　ｆｏｒ　Ｉｎｆｏｒｍａｔｉｏｎ　Ｉｎｔｅｒｃｈａｎｇｅ）コードに変換する。

【００９４】次いで、クライアント１００がクライアント２００に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【００９５】権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでいてもよい）。

【００９６】● オブジェクト名の列
● メソッド名の列
● 有効期間開始日時
● 有効期間終了日時

【００９７】ここで、クライアント１００が、クライアント２００にアクセス権限を委譲するために、以下の権限情報文字列ｃａｐａｂｉｌｉｔｙ１を生成したとする。

【００９８】

【数２】

((/object391),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)

【００９９】上記の権限情報文字列は、”／ｏｂｊｅｃｔ３９１”で表されるＨＴＴＰオブジェクト３９１に対して、ＧＥＴメソッドのみを、世界時で１９９９年の４月２４日１０時０分４８秒から１９９９年の４月２５日１０時０分４８秒の間だけ実行することのできる権利を表している。

【０１００】本実施例では、クライアント２００は、ＨＴＴＰサーバ３００のみを対象とするため、オブジェクト名としてサーバ内での名前のみを取り扱う。但し、一般には、オブジェクト名としてＵＲＬ等のグローバルな識別子を用いることにより、複数のＨＴＴＰサーバのオブジェクトを識別するようにしてもよい。

【０１０１】次いで、クライアント１００は、この権限情報ｃａｐａｂｉｌｉｔｙ１に対して秘密情報としてのｐａｓｓｗｏｒｄ１を文字列後方からビット連結して、一方向性関数ＭＤ（Ｍｅｓｓａｇｅ　Ｄｉｇｅｓｔ）を適用して、以下に示す保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を生成する。

【０１０２】

【数３】

MD(((/object),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)+password1)

【０１０３】一方向性関数ＭＤは、逆関数を求めることが極めて困難な関数であり、該関数ＭＤを適用する前の引数の値を推測することを不可能にする作用がある。したがって、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者が保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を勝手に改竄することはできない。

【０１０４】次いで、クライアント１００は、少なくとも、

● ｕｓｅｒｉｄ１（ユーザ情報）
● ｃａｐａｂｉｌｉｔｙ１（権限情報）
● ｃａｐａｂｉｌｉｔｙＭＤ１（保護化権限情報）

からなる情報を、他のクライアント２００に対して送信する。

【０１０５】既に述べたように、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者（クライアント２００を含む）は、保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を改竄することができない。したがって、クライアント１００は、他のクライアント２００に対して、ＨＴＴＰオブジェクト３９１に対するアクセス権限を安全に委譲することができる訳である。

【０１０６】他方、上記の情報を受信したクライアント２００は、クライアント１００の代理として、ＨＴＴＰオブジェクト３９１にアクセスするサービスを提供する。

【０１０７】クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするためには、少なくとも以下の情報を含む要求メッセージをＨＴＴＰサーバ３００に送信する。

【０１０８】● オブジェクト名
● メソッド名
● ユーザ名
● 権限情報
● 権限情報のメッセージ・ダイジェスト

【０１０９】本実施例では、以下に示すような要求メッ

セージを送信する。

【０１１０】

【数４】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｃａｐａｂｉｌｉｔｙ　ｃａｐ１

【０１１１】ここで、文字列ｃａｐ１は、ユーザ名と権限情報と権限情報のメッセージ・ダイジェストを”：”を用いて連結して、改行のないｂａｓｅ６４エンコード法（前述）でエンコードした文字列であるとする。

【０１１２】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、上記の要求メッセージ中のＡｕｔｈｏｒｉｚａｔｉｏｎフィールドから、以下の情報を取り出す。

- ユーザ情報
- 権限情報
- 権限情報のメッセージ・ダイジェスト

【０１１３】さらに、アクセス制御オブジェクト３０１は、取り出したこれらの情報を基に、以下に示すアルゴリズム１（図２を参照のこと）を実行することによって、クライアント２００がＨＴＴＰオブジェクト３９１に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【０１１４】アルゴリズム１：

（ステップ１）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出したユーザ名をｕｓｅｒｉｄとする。

【０１１５】（ステップ２）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出した権限情報をｃａｐａｂｉｌｉｔｙとする。

【０１１６】（ステップ３）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出した権限情報のメッセージ・ダイジェストをｃａｐａｂｉｌｉｔｙＭＤとする。

【０１１７】（ステップ４）ｕｓｅｒｉｄに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをｐａｓｓｗｏｒｄとする。他方、対応するパスワードが登録されていない場合は、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【０１１８】（ステップ５）権限情報ｃａｐａｂｉｌｉｔｙと秘密情報ｐａｓｓｗｏｒｄとをビット連結して、一方向性関数ＭＤを適用し、その結果をｄｉｇｅｓｔとする。

【０１１９】（ステップ６）ｄｉｇｅｓｔを、要求メッセージに含まれるｃａｐｃａｂｉｌｉｔｙＭＤと比較する。両者が等しくない場合には、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【０１２０】（ステップ７）Ａｕｔｏｒｉｚａｔｉｏｎフィールドから取り出されたｃａｐａｂｉｌｉｔｙを基に、クライアント２００が要求するメソッド（すなわちオブジェクトに対する処理内容）、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がｃａｐａｂｉｌｉｔｙで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【０１２１】アルゴリズム１を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１に対してＧＥＴメソッドを起動するメッセージを送信する。

【０１２２】図３には、本実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【０１２３】トランザクションの前提として、クライアント１００は、ＨＴＴＰサーバ３００に対するアクセスのアカウント（利用者資格）として、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。また、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…についてのユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」として保管している。

【０１２４】クライアント１００は、ＨＴＴＰオブジェクト３９１へのアクセスに関し、クライアント２００に与えてもよい権限情報ｃａｐａｂｉｌｉｔｙ１を作成する（Ｔｒ１）。

【０１２５】次いで、クライアント１００は、権限情報ｃａｐａｂｉｌｉｔｙ１の後ろに自己の秘密情報ｐａｓｓｗｏｒｄ１をビット連結したビット列に一方向性関数ＭＤを適用して、保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を作成する（Ｔｒ２）。

【０１２６】次いで、クライアント１００は、ユーザ情報ｕｓｅｒｉｄ１と権限情報ｃａｐａｂｉｌｉｔｙ１、及び保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１をクライアント２００に送信することで、アクセス権限を安全に委譲する（Ｔｒ３）。

【０１２７】アクセス権限を委譲されたクライアント２００は、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ４）。このアクセス要求には、クライアント１００のユーザ名ｕｓｅｒｉｄ１、権限情報ｃａｐａｂｉｌｉｔｙ１、及び保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を添付する。

【０１２８】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、クライアント２００のアクセス要求の正当性を検証する（Ｔｒ５）。該検証は、上述のアルゴリズム１（図２を参照のこと）が規定する処理手順に

従う。

【０１２９】検証結果が成功裡に終了した場合には、アクセス制御オブジェクト３０１はアクセス要求を受理する（Ｔｒ６）。より具体的には、要求されているメソッド（例えばＧＥＴメソッド）を起動する。

【０１３０】２．第２の実施形態次いで、本発明の第２の実施形態について説明する。第２の実施形態は、上記と同様、図１に示すような分散コンピューティング環境に適用可能である。第２の実施形態は、権限情報に一方向性関数を適用するとともに、「チャレンジ文字列」と呼ぶ１回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を採用するものである。

【０１３１】本実施形態において、ＨＴＴＰサーバ３００は、”ｈｔｔｐ：／／ｗｗｗ３００”なるＵＲＬで表されるものとする。また、ＨＴＴＰサーバ３００は、アクセス制御オブジェクト３０１と、参照番号３９１～３９Ｎで示されるＮ個のＨＴＴＰオブジェクトを所有している。各ＨＴＴＰオブジェクト３９１～３９Ｎはそれぞれ”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９１”，…，”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９Ｎ”なるＵＲＬで示されるものとする。

【０１３２】アクセス制御オブジェクト３０１は、各ＨＴＴＰオブジェクト３９１～３９Ｎに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１～３９Ｎと同じサーバ３００上に存在する必要は必ずしもなく、例えば別のサーバ（図示しない）上に存在してＨＴＴＰサーバ３００から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【０１３３】各クライアント１００…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント１００は、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。

【０１３４】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…に関するユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」（仮称）として保管している。

【０１３５】本実施形態では、ＨＴＴＰ１．１のＤｉｇｅｓｔ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ（基本認証法）に従ってＨＴＴＰメッセージが交換されるものとする。すなわち、クライアント１００は、以下に示すようなＨＴＴＰリクエストを発行することにより、ＨＴＴＰオブジェクト３９１のＧＥＴメソッドを起動することができる。

【０１３６】

【数５】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｄｉｇｅｓｔ　ｄｉｇｅｓｔ－ｒｅｓｐｏｎｓｅ

【０１３７】但し、”　ｄｉｇｅｓｔ－ｒｅｓｐｏｎｓｅ”は、ＨＴＴＰ１．１の規定に基づき生成された認証データである。

【０１３８】次いで、クライアント１００がクライアント２００に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【０１３９】権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでいてもよい）。

【０１４０】●　オブジェクト名の列
●　メソッド名の列
●　有効期間開始日時
●　有効期間終了日時

【０１４１】ここで、クライアント１００が、クライアント２００にアクセス権限を委譲するために、以下の権限情報文字列ｃａｐａｂｉｌｉｔｙ１を生成したとする。

【０１４２】

【数６】

((/object391),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)

【０１４３】上記の権限情報文字列は、”／ｏｂｊｅｃｔ３９１”で表されるＨＴＴＰオブジェクト３９１に対して、ＧＥＴメソッドのみを、世界時で１９９９年の４月２４日１０時０分４８秒から１９９９年の４月２５日１０時０分４８秒の間だけ実行することのできる権利を表している。

【０１４４】本実施例では、クライアント２００は、ＨＴＴＰサーバ３００のみを対象とするため、オブジェクト名としてサーバ内での名前のみを取り扱う。但し、一般には、オブジェクト名としてＵＲＬ等のグローバルな識別子を用いることにより、複数のＨＴＴＰサーバのオブジェクトを識別するようにしてもよい。

【０１４５】次いで、クライアント１００は、この権限情報ｃａｐａｂｉｌｉｔｙ１に対して秘密情報としてのｐａｓｓｗｏｒｄ１を文字列後方からビット連結して、一方向性関数ＭＤ（Ｍｅｓｓａｇｅ　Ｄｉｇｅｓｔ）を適用して、以下に示す保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を生成する。

【０１４６】

【数７】

MD(((/object),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)+password1)

【０１４７】一方向性関数ＭＤは、逆関数を求めることが極めて困難な関数であり、該関数ＭＤを適用する前の

引数の値を推測することを不可能にする作用がある。したがって、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者が保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を勝手に改竄することはできない。

【０１４８】次いで、クライアント１００は、少なくとも、

- ｕｓｅｒｉｄ１（ユーザ情報）
- ｃａｐａｂｉｌｉｔｙ１（権限情報）
- ｃａｐａｂｉｌｉｔｙＭＤ１（保護化権限情報）

からなる情報を、他のクライアント２００に対して送信する。

【０１４９】既に述べたように、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者（クライアント２００を含む）は、保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を改竄することができない。したがって、クライアント１００は、他のクライアント２００に対して、ＨＴＴＰオブジェクト３９１に対するアクセス権限を安全に委譲するこ＊とができる訳である。

【０１５０】他方、上記の情報を受信したクライアント２００は、クライアント１００の代理として、ＨＴＴＰオブジェクト３９１にアクセスするサービスを提供する。

【０１５１】クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするために、まず、以下に示すようなメッセージをＨＴＴＰサーバ３００に送信する。

【０１５２】

【数８】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ１．１

【０１５３】この場合、要求メッセージに認証データが含まれていないため、ＨＴＴＰサーバ３００は、以下のような応答メッセージをクライアント２００に送信する。

【０１５４】

【数９】

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Capability
        realm="private@fujixerox.co.jp",
        qop="auth,auth-int",
        nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
        opaque="a9f0171e7f40e415aa951069a403ecaf"
```

【０１５５】上記のうち、ｎｏｎｃｅに代入された文字列は、「チャレンジ・レスポンス認証」を行うための「チャレンジ文字列」である。「チャレンジ・レスポンス認証」とは、パスワード認証を安全に実施するための技術であり、１回限り使用するチャレンジ文字列を基に暗号化してメッセージ交換する方式である。

【０１５６】クライアント２００は、少なくともチャレ※ンジ文字列ｎｏｎｃｅと保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１とをビット連結した文字列のメッセージ・ダイジェストを計算する。この場合の例で言えば、以下の式を計算することになる。すなわち、

【０１５７】

【数１０】

$$MD(\text{“0a093dad98c11d0f600cfc7102dd2f0e8c”}+capabilityMD1)$$

【０１５８】また、その他の文字列もさらに連結するようにしてもよい。例えば、ＨＴＴＰ／１．１のように以下の式を計算してもよい。

★【０１５９】

【数１１】

★

```
MD(MD(useridl ":" private@fujixerox.co.jp ":" capabilityMD1)
+nonce ":" cnonce ":" qop ":" MD(GET ":" "/object391"))
```

【０１６０】上記の計算結果をｒｅｓｐｏｎｓｅ１とする。クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするためには、以下のような要求メッセージをＨＴＴＰサーバ３００に送信する。

【０１６１】

【数１２】

```
GET /object391 HTTP/1.1
Authorization: Capability
        username=userid1
        realm="provate@fujixerox.co.jp",
        nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
        uri="/object391",
        qop=auth,
        nc=00000001,
        cnonce="1bce22bf",
        capability=capability1,
        response=response1,
        opaque="a9f0171e7f40e415aa951069e403ecaf"
```

【０１６２】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、上記の要求メッセージ中のＡｕｔｈｏｒｉｚａｔｉｏｎフィールドから、以下の情報を取り出す。

- ユーザ情報
- 権限情報
- 権限情報のメッセージ・ダイジェスト

【０１６３】さらに、アクセス制御オブジェクト３０１は、取り出したこれらの情報を基に、以下に示すアルゴリズム２（図４を参照のこと）を実行することによって、クライアント２００がＨＴＴＰオブジェクト３９１に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【０１６４】アルゴリズム２：

（ステップ１１）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出したユーザ名をｕｓｅｒｉｄとする。

【０１６５】（ステップ１２）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出した権限情報をｃａｐａｂｉｌｉｔｙとする。

【０１６６】（ステップ１３）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出された権限情報のメッセージ・ダイジェストをｒｅｓｐｏｎｓｅとする。

【０１６７】（ステップ１４）クライアント２００に送信したメッセージ中のｎｏｎｃｅフィールドの値をｎｏｎｃｅ１とする。

【０１６８】（ステップ１５）ｕｓｅｒｉｄに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをｐａｓｓｗｏｒｄとする。他方、対応するパスワードが登録されていない場合は、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【０１６９】（ステップ１６）一方向性関数の値ＭＤ（ｎｏｎｃｅ１ ＋ ＭＤ（ｃａｐａｂｉｌｉｔｙ ＋ ｐａｓｓｗｏｒｄ））を計算し、計算結果をｄｉｇｅｓｔとする。

【０１７０】（ステップ１７）ｄｉｇｅｓｔをｒｅｓｐｏｎｓｅと比較する。両者が等しくない場合には、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【０１７１】（ステップ１８）ｃａｐａｂｉｌｉｔｙを基に、クライアント２００が要求するメソッド（すなわちオブジェクトに対する処理内容）、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がｃａｐａｂｉｌｉｔｙで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【０１７２】アルゴリズム２を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１に対してＧＥＴメソッドを起動するメッセージを送信する。

【０１７３】図５には、本実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【０１７４】トランザクションの前提として、クライアント１００は、ＨＴＴＰサーバ３００に対するアクセスのアカウント（利用者資格）として、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。また、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…についてのユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」として保管している。

【０１７５】クライアント１００は、ＨＴＴＰオブジェクト３９１へのアクセスに関し、クライアント２００に与えてもよい権限情報ｃａｐａｂｉｌｉｔｙ１を作成する（Ｔｒ１１）。

【０１７６】次いで、クライアント１００は、権限情報ｃａｐａｂｉｌｉｔｙ１の後ろに自己の秘密情報ｐａｓｓｗｏｒｄ１をビット連結したビット列に一方向性関数ＭＤを適用して、保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１を作成する（Ｔｒ１２）。

【０１７７】次いで、クライアント１００は、ユーザ情

報ｕｓｅｒｉｄ１と権限情報ｃａｐａｂｉｌｉｔｙ１、及び保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１をクライアント２００に送信することで、アクセス権限を安全に委譲する（Ｔｒ１３）。

【０１７８】アクセス権限を委譲されたクライアント２００は、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ１４）。

【０１７９】該アクセス要求には認証データが含まれていないので、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、チャレンジ文字列ｎｏｎｃｅを含んだ応答メッセージを送信する（Ｔｒ１５）。

【０１８０】クライアント２００は、チャレンジ文字列ｎｏｎｃｅと保護化権限情報ｃａｐａｂｉｌｉｔｙＭＤ１をビット連結したビット列のメッセージ・ダイジェストｒｅｓｐｏｎｓｅを生成する（Ｔｒ１６）。

【０１８１】そして、クライアント２００は、改めて、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ１７）。このアクセス要求には、クライアント１００のユーザ名ｕｓｅｒｉｄ１、権限情報ｃａｐａｂｉｌｉｔｙ１、及びメッセージ・ダイジェストｒｅｓｐｏｎｓｅを添付する。

【０１８２】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、クライアント２００のアクセス要求の正当性を検証する（Ｔｒ１８）。該検証は、上述のアルゴリズム１（図４を参照のこと）が規定する処理手順に従う。

【０１８３】検証結果が成功裡に終了した場合には、アクセス制御オブジェクト３０１はアクセス要求を受理する（Ｔｒ１９）。より具体的には、要求されているメソッド（例えばＧＥＴメソッド）を起動する。

【０１８４】３．第３の実施形態

次いで、本発明の第３の実施形態について説明する。第２の実施形態は、上記と同様、図１に示すような分散コンピューティング環境に適用可能である。但し、第３の実施形態は、上記の第１及び第２の実施形態とは相違し、権限情報に対して一方向性関数ではなく暗号化関数を適用することによってアクセス権限の安全な委譲を実現するものである。以下に示す例では、対称暗号鍵方式を用いた場合について説明するが、公開暗号鍵方式を用いても同様に本発明の効果を奏する点を充分理解されたい。

【０１８５】本実施形態において、ＨＴＴＰサーバ３００は、”ｈｔｔｐ：／／ｗｗｗ３００”なるＵＲＬで表されるものとする。また、ＨＴＴＰサーバ３００は、アクセス制御オブジェクト３０１と、参照番号３９１～３９Ｎで示されるＮ個のＨＴＴＰオブジェクトを所有している。各ＨＴＴＰオブジェクト３９１～３９Ｎはそれぞれ”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９１”，…，”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９Ｎ”なるＵＲＬで示されるものとする。

【０１８６】アクセス制御オブジェクト３０１は、各ＨＴＴＰオブジェクト３９１～３９Ｎに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１～３９Ｎと同じサーバ３００上に存在する必要は必ずしもなく、例えば別のホスト（図示しない）上に存在してＨＴＴＰサーバ３００から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【０１８７】各クライアント１００…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント１００は、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。

【０１８８】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…に関するユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」（仮称）として保管している。

【０１８９】また、アクセス制御オブジェクト３０１は、オブジェクトに対するアクセス権限を与えたクライアント・ユーザに対する「ケイパビリティ（ｃａｐａｂｉｌｉｔｙ）」を保持している。ケイパビリティとは、アクセス制御マトリックス（前述）を行毎すなわちユーザ毎に切り出した情報であり、ユーザが各ＨＴＴＰオブジェクトに対して許容された操作権限を記述した情報である。

【０１９０】各クライアント１００…は、ＨＴＴＰサーバ３００に対して自己のユーザ情報とパスワードを提示することによって、自身に与えられたケイパビリティすなわち権限情報の範囲内でＨＴＴＰオブジェクト３９１…に対するアクセス操作が許容される。

【０１９１】本実施形態では、ＨＴＴＰ１．１のＢａｓｉｃ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ（基本認証法）に従ってＨＴＴＰメッセージが交換されるものとする。すなわち、クライアント１００は、以下に示すようなＨＴＴＰリクエストを発行することにより、ＨＴＴＰオブジェクト３９１のＧＥＴメソッドを起動することができる。

【０１９２】

【数１３】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｂａｓｉｃ　ａｕｔｈ　ｄａｔａ

【０１９３】但し、”ａｕｔｈｄａｔａ”は、ユーザ情報とパスワードを連結したビット列ｕｓｅｒｉｄ１：ｐａｓｓｗｏｒｄ１を、改行のないｂａｓｅ６４エンコード法（前述）でエンコードした文字列である。

【０１９４】次いで、クライアント１００がクライアン

ト２００に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【０１９５】権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでいてもよい）。

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時
- 有効期間終了日時

【０１９６】ここで、クライアント１００が、クライアント２００にアクセス権限を委譲するために、以下の権限情報文字列ｃａｐａｂｉｌｉｔｙ１を生成したとする。

【０１９７】

【数１４】

((/object391),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)

【０１９８】上記の権限情報文字列は、”／ｏｂｊｅｃｔ３９１”で表されるＨＴＴＰオブジェクト３９１に対して、ＧＥＴメソッドのみを、世界時で１９９９年の４月２４日１０時０分４８秒から１９９９年の４月２５日１０時０分４８秒の間だけ実行することのできる権利を表している。

【０１９９】本実施例では、クライアント２００は、ＨＴＴＰサーバ３００のみを対象とするため、オブジェクト名としてサーバ内での名前のみを取り扱う。但し、一般には、オブジェクト名としてＵＲＬ等のグローバルな識別子を用いることにより、複数のＨＴＴＰサーバのオブジェクトを識別するようにしてもよい。

＃ｄ

次いで、クライアント１００は、自己の秘密情報ｐａｓｓｗｏｒｄ１を用いてこの権限情報ｃａｐａｂｉｌｉｔｙ１を暗号化して、以下に示す保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を生成する。

【０２００】

【数１５】

CRYPT(((/object),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT),password1)

【０２０１】上記のうち暗号化関数ＣＲＹＰＴは、例えば、ＤＥＳ（Ｄａｔａ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）やＲＣ２、ＲＣ４、ＲＣ５などの任意の対称鍵暗号系を用いることができる。ＤＥＳとは、米国政府が標準暗号として認定した共通鍵暗号方式のことである。鍵は５６ビットの固定長（８ビットのパリティ・ビットを含めて６４ビット）であり、まず５６ビット長の鍵から４８ビット長の鍵を１６個作り、その鍵を使って６４ビット長に分けたデータ・ブロックを１６回だけ攪拌処理する。

【０２０２】また、ＲＣ２、ＲＣ４及びＲＣ５は、米ＲＳＡデータ・セキュリティ社が開発した共通鍵暗号方式である。このうち、ＲＣ２はブロック毎にデータを暗号化するブロック暗号であり、暗号鍵の長さを可変にできる。ＲＣ４は「ストリーム型」と呼ぶ暗号方式を採用する。ＲＣ５は鍵長、データ・ブロック長ともに可変の暗号方式である。

【０２０３】元の権限情報に対してこのような暗号化関数ＣＲＹＰＴを適用することによって、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者が保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を勝手に改竄することはできない。

【０２０４】次いで、クライアント１００は、少なくとも、

- ｕｓｅｒｉｄ１（ユーザ情報）
- ｃａｐａｂｉｌｉｔｙＣＲ１（保護化権限情報）

からなる情報を、他のクライアント２００に対して送信する。

【０２０５】既に述べたように、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者（クライアント２００を含む）は、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を改竄することができない。したがって、クライアント１００は、他のクライアント２００に対して、ＨＴＴＰオブジェクト３９１に対するアクセス権限を安全に委譲することができる訳である。

【０２０６】他方、上記の情報を受信したクライアント２００は、クライアント１００の代理として、ＨＴＴＰオブジェクト３９１にアクセスするサービスを提供する。

【０２０７】クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするためには、少なくとも以下の情報を含む要求メッセージをＨＴＴＰサーバ３００に送信する。

【０２０８】
- オブジェクト名
- メソッド名
- ユーザ名
- 暗号化された権限情報

【０２０９】本実施例では、以下に示すような要求メッセージを送信する。

【０２１０】

【数１６】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｃａｐａｂｉｌｉｔｙ　ｃａｐ１

【０２１１】ここで、文字列ｃａｐ１は、ユーザ名と暗号化された権限情報とを”：”を用いて連結して改行のないｂａｓｅ６４エンコード法（前述）でエンコードした文字列であるとする。

【０２１２】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、上記の要求メッセージ中のＡｕｔｈｏｒｉｚａｔｉｏｎフィールドから、以下の情報を取り

出す。

● ユーザ情報

● 暗号化された権限情報

【０２１３】さらに、アクセス制御オブジェクト３０１は、取り出したこれらの情報を基に、以下に示すアルゴリズム３（図６を参照のこと）を実行することによって、クライアント２００がＨＴＴＰオブジェクト３９１に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【０２１４】アルゴリズム３：

（ステップ２１）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出したユーザ名をｕｓｅｒｉｄとする。

【０２１５】（ステップ２２）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出した暗号化された権限情報をｃａｐａｂｉｌｉｔｙＣＲとする。

【０２１６】（ステップ２３）ｕｓｅｒｉｄに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをｐａｓｓｗｏｒｄとする。他方、対応するパスワードが登録されていない場合は、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【０２１７】（ステップ２４）暗号化された権限情報ｃａｐａｂｉｌｉｔｙＣＲを、秘密情報ｐａｓｓｗｏｒｄを用いて復号化する。復号化結果をｃａｐａｂｉｌｉｔｙとする。

【０２１８】（ステップ２５）復号化された権限情報ｃａｐａｂｉｌｉｔｙが正しい文法規則に則っているか否かを判断する。正しくない場合には、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。正しい場合には次ステップへ進む。

【０２１９】（ステップ２６）復号化されたｃａｐａｂｉｌｉｔｙを基に、クライアント２００が要求するメソッド（すなわちオブジェクトに対する処理内容）、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がｃａｐａｂｉｌｉｔｙで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【０２２０】アルゴリズム３を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１に対してＧＥＴメソッドを起動するメッセージを送信する。

【０２２１】図７には、本実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【０２２２】トランザクションの前提として、クライアント１００は、ＨＴＴＰサーバ３００に対するアクセスのアカウント（利用者資格）として、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。また、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…についてのユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」として保管している。

【０２２３】クライアント１００は、ＨＴＴＰオブジェクト３９１へのアクセスに関し、クライアント２００に与えてもよい権限情報ｃａｐａｂｉｌｉｔｙ１を作成する（Ｔｒ２１）。

【０２２４】次いで、クライアント１００は、自己の秘密情報ｐａｓｓｗｏｒｄ１を用いて権限情報ｃａｐａｂｉｌｉｔｙ１を暗号化して、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を作成する（Ｔｒ２２）。

【０２２５】次いで、クライアント１００は、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１をクライアント２００に送信することで、アクセス権限を安全に委譲する（Ｔｒ２３）。

【０２２６】アクセス権限を委譲されたクライアント２００は、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ２４）。このアクセス要求には、クライアント１００のユーザ名ｕｓｅｒｉｄ１と保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を添付する。

【０２２７】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、クライアント２００のアクセス要求の正当性を検証する（Ｔｒ２５）。該検証は、上述のアルゴリズム３（図６を参照のこと）が規定する処理手順に従う。

【０２２８】検証結果が成功裡に終了した場合には、アクセス制御オブジェクト３０１はアクセス要求を受理する（Ｔｒ２６）。より具体的には、要求されているメソッド（例えばＧＥＴメソッド）を起動する。

【０２２９】４．第４の実施形態

次いで、本発明の第４の実施形態について説明する。第４の実施形態は、上記と同様、図１に示すような分散コンピューティング環境に適用可能である。第４の実施形態は、権限情報を暗号化するとともに、クライアント－ＨＴＴＰサーバ間のアクセス要求の際に「チャレンジ・レスポンス認証」（前述）を用いることによってアクセス権限の安全な委譲を実現するものである。

【０２３０】本実施形態において、ＨＴＴＰサーバ３００は、”ｈｔｔｐ：／／ｗｗｗ３００”なるＵＲＬで表されるものとする。また、ＨＴＴＰサーバ３００は、アクセス制御オブジェクト３０１と、参照番号３９１～３９Ｎで示されるＮ個のＨＴＴＰオブジェクトを所有している。各ＨＴＴＰオブジェクト３９１～３９Ｎはそれぞれ”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅｃｔ３９１”，…，”　ｈｔｔｐ：／／ｗｗｗ３００／ｏｂｊｅ

ｃｔ３９Ｎ”なるＵＲＬで示されるものとする。

【０２３１】アクセス制御オブジェクト３０１は、各ＨＴＴＰオブジェクト３９１～３９Ｎに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１～３９Ｎと同じサーバ３００上に存在する必要は必ずしもなく、例えば別のサーバ（図示しない）上に存在してＨＴＴＰサーバ３００から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【０２３２】各クライアント１００…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント１００は、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。

【０２３３】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…に関するユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」（仮称）として保管している。

【０２３４】また、アクセス制御オブジェクト３０１は、オブジェクトに対するアクセス権限を与えたクライアント・ユーザに対する「ケイパビリティ（ｃａｐａｂｉｌｉｔｙ）」を保持している。ケイパビリティとは、アクセス制御マトリックス（前述）を行毎すなわちユーザ毎に切り出した情報であり、ユーザが各ＨＴＴＰオブジェクトに対して許容された操作権限を記述した情報である。

【０２３５】各クライアント１００…は、ＨＴＴＰサーバ３００に対して自己のユーザ情報とパスワードを提示することによって、自身に与えられたケイパビリティすなわち権限情報の範囲内でＨＴＴＰオブジェクト３９１…に対するアクセス操作が許容される。

【０２３６】本実施形態では、ＨＴＴＰ１．１のＤｉｇｅｓｔ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ（基本認証法）に従ってＨＴＴＰメッセージが交換されるものとする。すなわち、クライアント１００は、以下に示すようなＨＴＴＰリクエストを発行することにより、ＨＴＴＰオブジェクト３９１のＧＥＴメソッドを起動することができる。

【０２３７】

【数１７】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ／１．１
Ａｕｔｈｏｒｉｚａｔｉｏｎ：　Ｄｉｇｅｓｔ　ｄｉｇｅｓｔ－ｒｅｓｐｏｎｓｅ

【０２３８】但し、”　ｄｉｇｅｓｔ－ｒｅｓｐｏｎｓｅ”は、ＨＴＴＰ１．１の規定に基づき生成された認証データである。

【０２３９】次いで、クライアント１００がクライアント２００に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【０２４０】権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでいてもよい）。

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時
- 有効期間終了日時

【０２４１】ここで、クライアント１００が、クライアント２００にアクセス権限を委譲するために、以下の権限情報文字列ｃａｐａｂｉｌｉｔｙ１を生成したとする。

【０２４２】

【数１８】

((/object391),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT)

【０２４３】上記の権限情報文字列は、”／ｏｂｊｅｃｔ３９１”で表されるＨＴＴＰオブジェクト３９１に対して、ＧＥＴメソッドのみを、世界時で１９９９年の４月２４日１０時０分４８秒から１９９９年の４月２５日１０時０分４８秒の間だけ実行することのできる権利を表している。

【０２４４】次いで、クライアント１００は、自己の秘密情報ｐａｓｓｗｏｒｄ１を用いてこの権限情報ｃａｐａｂｉｌｉｔｙ１を暗号化して、以下に示す保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を生成する。

【０２４５】

【数１９】

CRYPT(((/object),(GET),Apr:24:10:00:48:1999:GMT,Apr:25:10:00:48:1999:GMT),password1)

【０２４６】上記のうち暗号化関数ＣＲＹＰＴは、例えば、ＤＥＳ（Ｄａｔａ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）やＲＣ２、ＲＣ４、ＲＣ５などの任意の対称鍵暗号系（前述）を用いることができる。元の権限情報に対してこのような暗号化を適用することによって、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者が保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を勝手に改竄することはできない。

【０２４７】次いで、クライアント１００は、少なくとも、

- ｕｓｅｒｉｄ１（ユーザ情報）
- ｃａｐａｂｉｌｉｔｙ１（権限情報）
- ｃａｐａｂｉｌｉｔｙＣＲ１（保護化権限情報）

からなる情報を、他のクライアント２００に対して送信する。

【０２４８】既に述べたように、秘密情報ｐａｓｓｗｏｒｄ１を知らない第３者（クライアント２００を含む）は、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を改竄

43

することができない。したがって、クライアント１００は、他のクライアント２００に対して、ＨＴＴＰオブジェクト３９１に対するアクセス権限を安全に委譲することができる訳である。

【０２４９】他方、上記の情報を受信したクライアント２００は、クライアント１００の代理として、ＨＴＴＰオブジェクト３９１にアクセスするサービスを提供する。

【０２５０】クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするために、まず、以下に示すよ＊10

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Capability
        realm="private@fujixerox.co.jp",
        qop="auth,auth-int",
        nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
        opaque="a9f0171e7f40e415aa951069a403ecaf"
```

【０２５４】上記のうち、ｎｏｎｃｅに代入された文字列は、「チャレンジ・レスポンス認証」を行うための「チャレンジ文字列」（前述）である。

【０２５５】クライアント２００は、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を用いてチャレンジ文字列ｎ※

$$CRYPT(\text{“0a093dad98c11d0f600cfc7102dd2f0e8c”}, capabilityCR1)$$

【０２５７】また、その他の文字列もさらに連結するようにしてもよい。例えば、ＨＴＴＰ／１．１のように以下の式を計算してもよい。★

```
CRYPT(CRYPT(userid1 ":" private@fujixerox.co.jp ":"
capabilityCR1)+nonce ":" cnonce ":" qop ":"
CRYPT(GET ":" "/object391"))
```

【０２５９】上記の計算結果をｒｅｓｐｏｎｓｅ１とする。クライアント２００は、ＨＴＴＰオブジェクト３９１にアクセスするためには、以下のような要求メッセー☆

```
GET /object391 HTTP/1.1
Authorization: Capability
        username=userid1
        realm="provate@fujixerox.co.jp",
        nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
        uri="/object391",
        qop=auth,
        nc=00000001,
        cnonce="1bce22bf",
        capability=capability1,
        response=response1,
        opaque="a9f0171e7f40e415aa951069a403ecaf"
```

【０２６１】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、上記の要求メッセージ中のＡｕｔｈｏｒｉｚａｔｉｏｎフィールドから、以下の情報を取り出す。

- ユーザ情報

44

＊うなメッセージをＨＴＴＰサーバ３００に送信する。

【０２５１】

【数２０】

ＧＥＴ　／ｏｂｊｅｃｔ３９１　ＨＴＴＰ１．１

【０２５２】この場合、要求メッセージに認証データが含まれていないため、ＨＴＴＰサーバ３００は、以下のような応答メッセージをクライアント２００に送信する。

【０２５３】

【数２１】

※ｏｎｃｅを暗号化した値を計算する。この場合の例で言えば、以下の式を計算することになる。すなわち、

【０２５６】

【数２２】

★【０２５８】

【数２３】

☆ジをＨＴＴＰサーバ３００に送信する。

【０２６０】

【数２４】

- 権限情報
- 権限情報を暗号化した値

【０２６２】さらに、アクセス制御オブジェクト３０１は、取り出したこれらの情報を基に、以下に示すアルゴリズム４（図８を参照のこと）を実行することによっ

て、クライアント２００がＨＴＴＰオブジェクト３９１に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【０２６３】アルゴリズム４：

（ステップ３１）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出したユーザ名をｕｓｅｒｉｄとする。

【０２６４】（ステップ３２）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出した権限情報をｃａｐａｂｉｌｉｔｙとする。

【０２６５】（ステップ３３）Ａｕｔｈｏｒｉｚａｔｉｏｎフィールドから抽出された権限情報の暗号化値をｒｅｓｐｏｎｓｅとする。

【０２６６】（ステップ３４）クライアント２００に送信したメッセージ中のｎｏｎｃｅフィールドの値をｎｏｎｃｅ１とする。

【０２６７】（ステップ３５）ｕｓｅｒｉｄに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをｐａｓｓｗｏｒｄとする。他方、対応するパスワードが登録されていない場合は、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【０２６８】（ステップ３６）暗号化関数の値ＣＲＹＰＴ（ｎｏｎｃｅ１，ＣＲＹＰＴ（ｃａｐａｂｉｌｉｔｙ，　ｐａｓｓｗｏｒｄ））を計算し、計算結果をｄｉｇｅｓｔとする。

【０２６９】（ステップ３７）ｄｉｇｅｓｔをｒｅｓｐｏｎｓｅと比較する。両者が等しくない場合には、クライアント２００からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【０２７０】（ステップ３８）ｃａｐａｂｉｌｉｔｙを基に、クライアント２００が要求するメソッド（すなわちオブジェクトに対する処理内容）、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がｃａｐａｂｉｌｉｔｙで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【０２７１】アルゴリズム４を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト３０１は、ＨＴＴＰオブジェクト３９１に対してＧＥＴメソッドを起動するメッセージを送信する。

【０２７２】図９には、本実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【０２７３】トランザクションの前提として、クライアント１００は、ＨＴＴＰサーバ３００に対するアクセスのアカウント（利用者資格）として、ユーザ情報”ｕｓｅｒｉｄ１”とパスワード”ｐａｓｓｗｏｒｄ１”を保持しているものとする。また、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、各クライアント１００…についてのユーザ情報とパスワードの組み合わせ（ｕｓｅｒｉｄ１，ｐａｓｓｗｏｒｄ１），（ｕｓｅｒｉｄ２，ｐａｓｓｗｏｒｄ２）…を「パスワード管理表」として保管している。

【０２７４】クライアント１００は、ＨＴＴＰオブジェクト３９１へのアクセスに関し、クライアント２００に与えてもよい権限情報ｃａｐａｂｉｌｉｔｙ１を作成する（Ｔｒ３１）。

【０２７５】次いで、クライアント１００は、自己の秘密情報ｐａｓｓｗｏｒｄ１を用いて権限情報ｃａｐａｂｉｌｉｔｙ１を暗号化して、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲＹＰＴ１を作成する（Ｔｒ３２）。

【０２７６】次いで、クライアント１００は、ユーザ情報ｕｓｅｒｉｄ１と権限情報ｃａｐａｂｉｌｉｔｙ１、及び保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲＹＰＴ１をクライアント２００に送信することで、アクセス権限を委譲する（Ｔｒ３３）。

【０２７７】アクセス権限を委譲されたクライアント２００は、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ３４）。

【０２７８】該アクセス要求には認証データが含まれていないので、ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、チャレンジ文字列ｎｏｎｃｅを含んだ応答メッセージを送信する（Ｔｒ３５）。

【０２７９】クライアント２００は、保護化権限情報ｃａｐａｂｉｌｉｔｙＣＲ１を用いてチャレンジ文字列ｎｏｎｃｅを暗号化して、ｒｅｓｐｏｎｓｅを生成する（Ｔｒ３６）。

【０２８０】そして、クライアント２００は、改めて、ＨＴＴＰサーバ３００に対してＨＴＴＰオブジェクト３９１へのアクセス要求を送信する（Ｔｒ３７）。このアクセス要求には、クライアント１００のユーザ名ｕｓｅｒｉｄ１、権限情報ｃａｐａｂｉｌｉｔｙ１、及びｒｅｓｐｏｎｓｅを添付する。

【０２８１】ＨＴＴＰサーバ３００のアクセス制御オブジェクト３０１は、クライアント２００のアクセス要求の正当性を検証する（Ｔｒ３８）。該検証は、上述のアルゴリズム１（図８を参照のこと）が規定する処理手順に従う。

【０２８２】検証結果が成功裡に終了した場合には、アクセス制御オブジェクト３０１はアクセス要求を受理する（Ｔｒ３９）。より具体的には、要求されているメソッド（例えばＧＥＴメソッド）を起動する。

【０２８３】５．保護化権限情報の安全な保存方法

上述した第１～第４の実施形態によれば、クライアント１００は、他のクライアント２００に対してＨＴＴＰオブジェクトへのアクセス権限を委譲するに際し、自らの

パスワードｐａｓｓｗｏｒｄ１をそのまま渡したり、クライアント２００に与える権限情報ｃａｐａｂｉｌｉｔｙ１そのものを送信することはせず、権限情報ｃａｐａｂｉｌｉｔｙ１に対して一方向性関数ＭＤや暗号化関数ＣＲＹＰＴを適用して、権限情報を不正に改竄できないような「保護化権限情報」の形態にしてから委譲するようにした。

【０２８４】この結果、権限情報の内容を勝手に書き換えてＨＴＴＰオブジェクトに不正アクセスされる機会（例えば有効期限を無断で延長したり、有効使用回数を増やしたり、オブジェクトへの操作権限を強化したすること）を、好適に排除することができる。

【０２８５】しかしながら、保護された形式であれ、権限情報を複製することで、権限を正当に譲受していない者さえＨＴＴＰオブジェクトにアクセスすることが可能になってしまう。

【０２８６】例えば、図１０に示すように、クライアント１００から正当にアクセス権限を譲受したクライアント２００が、自らは権限情報を正当にのみ使用していたとしても、保護化権限情報を格納した２次記憶装置（例えばハード・ディスク装置）２０１が攻撃を受けることで、保護化権限情報は漏洩し、悪用を許してしまいかねない。メモリ空間すなわち揮発的なメモリ上にロードされた保護化権限情報をプロテクトすることは比較的容易であるが、ファイル空間すなわちローカル・ディスク上の格納された情報をプロテクトすることは容易ではない。特に、分散コンピューティング環境では、リモートのホスト装置間で互いのローカル・ディスクはトランスパレントな状態なので、攻撃を受け易い。

【０２８７】以下では、クライアント２００が委譲された保護化権限情報をローカル・ディスクに安全且つ不揮発的に格納する方法について説明する。

【０２８８】５－１．方法１

まず、クライアント１００は、他のクライアント２００に保護化権限情報を送信する。

【０２８９】次いで、クライアント１００は、クライアント２００に対して暗号化鍵を送信する。

【０２９０】クライアント２００は、保護化権限情報を自身のローカル・ディスク２０１に格納するときには、受信した暗号化鍵を用いて暗号化してから行う。この結果、不正クライアントがローカル・ディスク２０１に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【０２９１】クライアント２００がＨＴＴＰオブジェクト３９１へのアクセス要求を行うときには、ローカル・ディスク２０１から保護化権限情報を取り出すとともに、暗号化鍵で復号化して用いればよい。保護化権限情報はクライアント２００のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【０２９２】ＨＴＴＰサーバ３００に対するアクセス要求及び認証の処理手順は上記に従う。

【０２９３】５－２．方法２

まず、クライアント１００は、暗号化鍵を用いて保護化権限情報を暗号化して、２重に保護化された権限情報を生成する。

【０２９４】次いで、クライアント１００は、他のクライアント２００に、２重に保護化された権限情報を送信する。

【０２９５】そして、クライアント２００は、受け取った保護化権限情報をローカル・ディスク２０１にそのまま格納する。この保護化権限情報は二重に保護され、そのままアクセス要求に使用することはできない。したがって、不正クライアントがローカル・ディスク２０１に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【０２９６】クライアント２００がオブジェクトのアクセス要求を発行したいときには、２重に保護化された権限情報を復号化するための復号化鍵を、クライアント１００から受け取ればよい。

【０２９７】そして、クライアント２００は、受け取った復号化鍵を用いて２重に保護化された権限情報を復号化することで、利用可能な形式の保護化権限情報を得ることができる。保護化権限情報はクライアント２００のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【０２９８】ＨＴＴＰサーバ３００に対するアクセス要求及び認証の処理手順は上記に従う。

【０２９９】なお、暗号化鍵と復号化鍵は、対称鍵暗号方式における同一鍵であっても、あるいは公開鍵暗号方式における秘密鍵と公開鍵の組み合わせであってもよい。

【０３００】５－３．方法３

まず、クライアント１００は、他のクライアント２００に保護化権限情報を送信する。

【０３０１】クライアント２００は、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを用意しておく。暗号化鍵と復号化鍵は、公開鍵暗号方式における秘密鍵と公開鍵の組み合わせでよい。

【０３０２】クライアント２００は、復号化鍵をクライアント１００に送信する。復号化鍵をローカル・ディスク２０１に格納してしまうと、不正クライアントがローカル・ディスク２０１に攻撃を仕掛けて、外部に漏洩する危険がある。このため、クライアント２００は、クライアント１００に復号化鍵を送信した後、直ちに復号化鍵を廃棄しておくことが好ましい。

【０３０３】そして、クライアント２００は、保護化権限情報をさらに暗号化鍵で暗号化してから、ローカル・ディスクに格納する。この結果、不正クライアントがロ

ーカル・ディスク２０１に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【０３０４】クライアント２００は、オブジェクトのアクセス要求を発行したいときには、２重に保護化された権限情報をローカル・ディスク２０１から取り出すとともに、復号化鍵を用いてこれを復号化して、元の保護化権限情報を再現すればよい。但し、クライアント２００が復号化鍵を廃棄してしまっている場合には、クライアント１００から復号化鍵を改めて取り寄せればよい。保護化権限情報はクライアント２００のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【０３０５】ＨＴＴＰサーバ３００に対するアクセス要求及び認証の処理手順は上記に従う。

【０３０６】５－４．方法４

まず、クライアント１００は、他のクライアント２００に保護化権限情報を送信する。

【０３０７】次いで、クライアント２００は、クライアント１００に「チャレンジ・レスポンス認証」（前述）を行うためのチャレンジ文字列を送信する。

【０３０８】次いで、クライアント１００は、受信したチャレンジ文字列と所定の秘密情報に対して所定の演算処理を適用して暗号鍵を生成する。そして、クライアント１００は、生成した暗号鍵をクライアント２００に送信する。

【０３０９】これに対し、クライアント２００は、受信した暗号鍵を用いて保護化権限情報を暗号化してから、ローカル・ディスク２０１に格納する。この結果、不正クライアントがローカル・ディスク２０１に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【０３１０】クライアント２００がＨＴＴＰオブジェクト３９１へのアクセス要求を行うときには、ローカル・ディスク２０１から保護化権限情報を取り出すとともに、暗号化鍵で復号化して用いればよい。保護化権限情報はクライアント２００のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【０３１１】ＨＴＴＰサーバ３００に対するアクセス要求及び認証の処理手順は上記に従う。

【０３１２】なお、上記の各方法では、保護化権限情報を安全且つ不揮発的にローカル・ディスク２０１に格納する場合を例にとって説明したが、格納する情報は保護化権限情報に限定されず、他の秘密性の高い情報（例えばユーザのパスワードなど）の格納にも応用することができる。

【０３１３】［追補］以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０３１４】なお、ＵＲＬ（Ｕｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒ）については、例えばＲＦＣ（Ｒｅｑｕｅｓｔ Ｆｏｒ Ｃｏｍｍｅｎｔｓ）１７３８やＲＦＣ１８０８などに記述されている。また、ＨＴＭＬ（Ｈｙｐｅｒ ＴｅｘｔＭａｒｋｕｐ Ｌａｎｇｕａｇｅ）については、例えばＲＦＣ１８６６に記述されている。また、ＨＴＴＰ（Ｈｙｐｅｒ Ｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ）については、例えばＲＦＣ１９４５やＲＦＣ２０６８に記述されている。

【０３１５】

【発明の効果】以上詳記したように、本発明によれば、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することができる。

【０３１６】また、本発明によれば、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」を、ホスト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することができる。

【０３１７】また、本発明によれば、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、他のサブジェクトに安全に委譲することができる、優れたアクセス権限委譲方法を提供することができる。

【０３１８】また、本発明によれば、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、生成されたケイパビリティをオブジェクトの管理者が安全に検査することができる、優れたアクセス権限委譲方法を提供することができる。

【図面の簡単な説明】

【図１】本発明の実施形態に係る分散コンピューティング環境を模式的に示した図である。

【図２】本発明の第１の実施形態において、アクセス権限オブジェクト３０１がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図３】本発明の第１の実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解していたものである。

【図４】本発明の第２の実施形態において、アクセス権限オブジェクト３０１がアクセスの正当性を判断するた

51

めの処理手順を示したフローチャートである。

【図５】本発明の第２の実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解していたものである。

【図６】本発明の第３の実施形態において、アクセス権限オブジェクト３０１がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図７】本発明の第３の実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解していたものである。

【図８】本発明の第４の実施形態において、アクセス権限オブジェクト３０１がアクセスの正当性を判断するた＊

52

＊めの処理手順を示したフローチャートである。

【図９】本発明の第４の実施形態において、クライアント１００、クライアント２００及びＨＴＴＰサーバ３００間で実行されるトランザクションを図解していたものである。

【図１０】本発明の実施形態に係る分散コンピューティング環境を模式的に示した図であり、より具体的にはクライアント２００の２次記憶装置に格納した保護化権限情報が不正利用される様子を図解したものである。

【符号の説明】

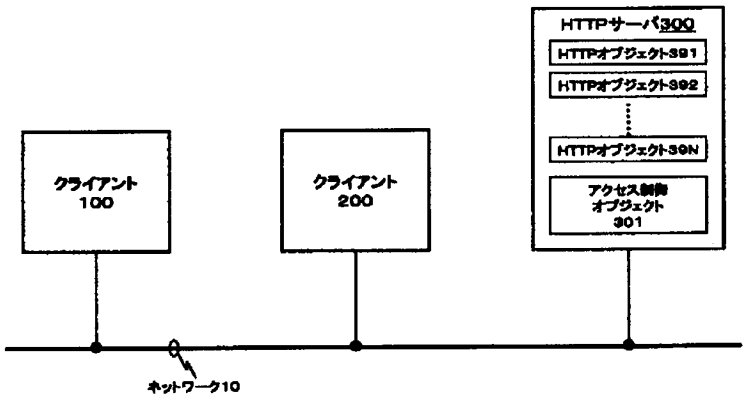
１００，２００…クライアント

３００…ＨＴＴＰサーバ

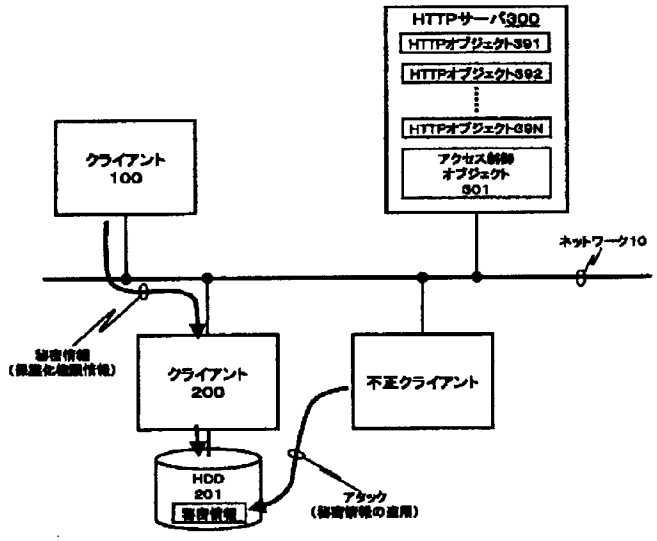
３０１…アクセス制御オブジェクト

３９１～３９Ｎ…ＨＴＴＰオブジェクト

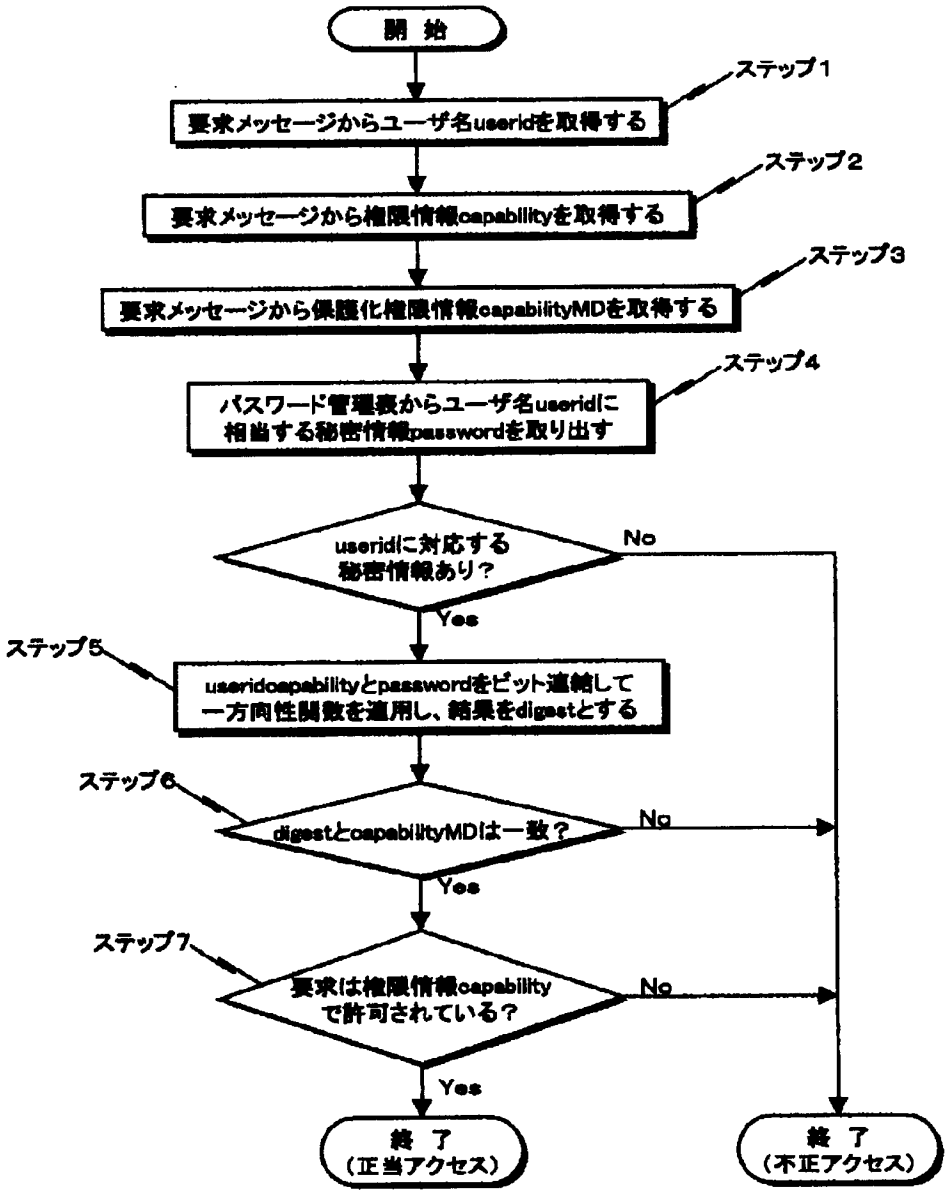
【図１】



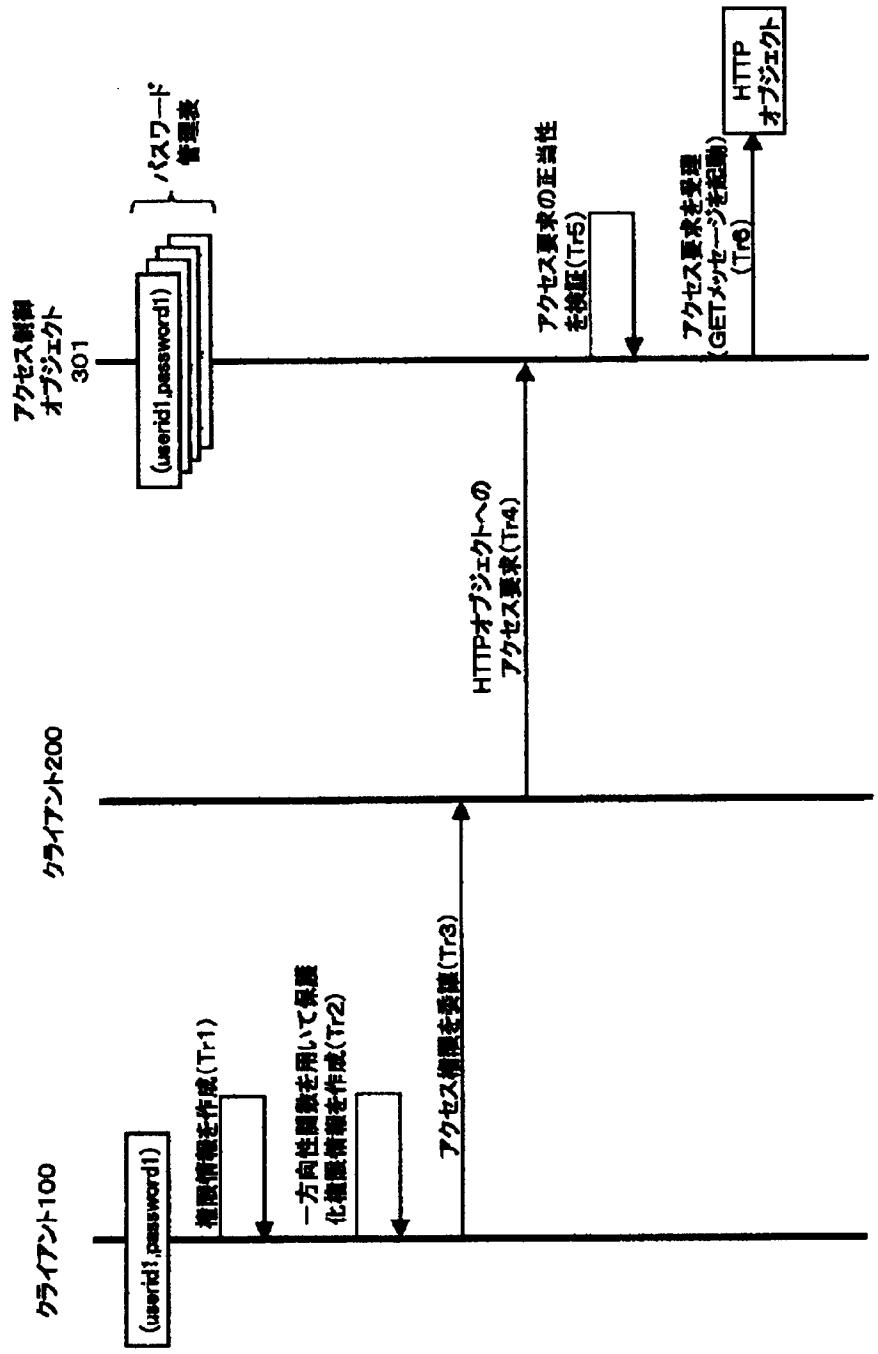
【図１０】



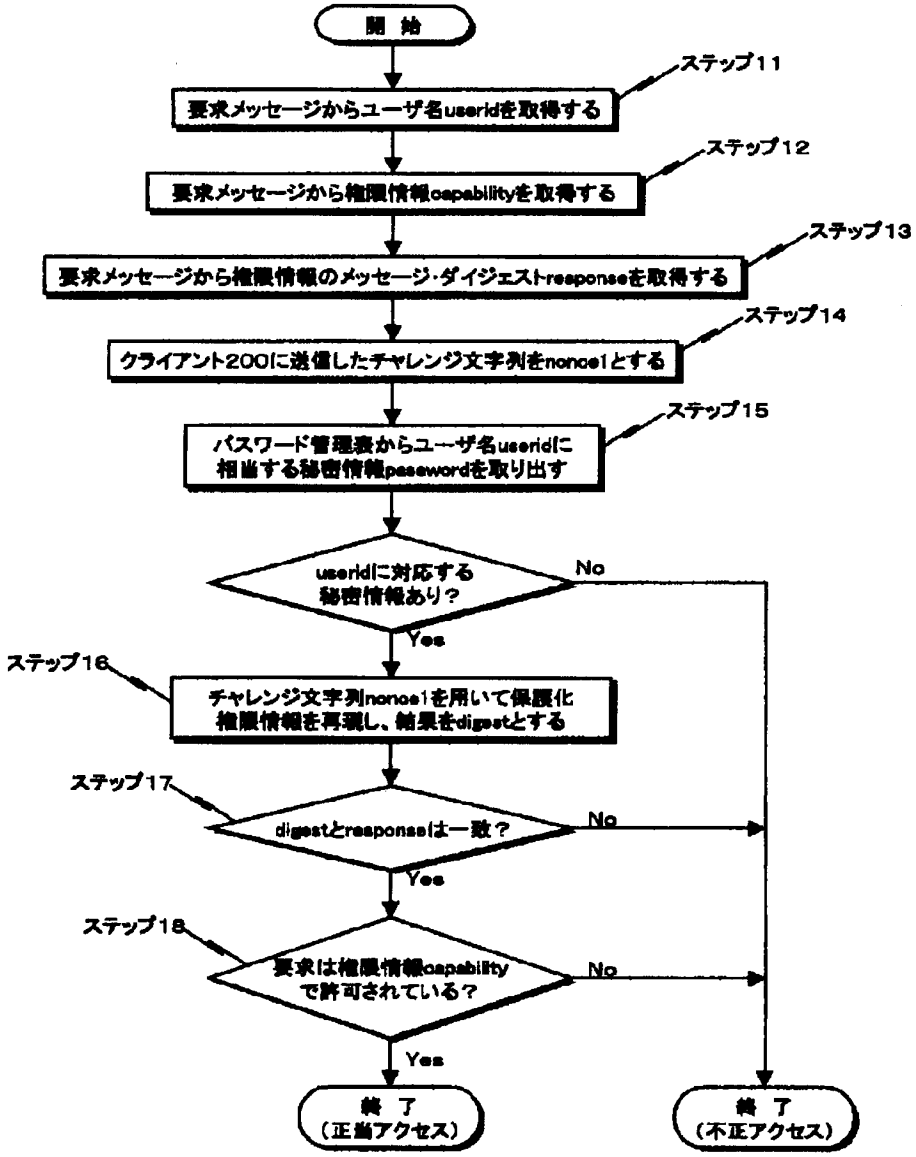
【図２】


開 始
ステップ1
要求メッセージからユーザ名useridを取得する
ステップ2
要求メッセージから権限情報capabilityを取得する
ステップ3
要求メッセージから保護化権限情報capabilityMDを取得する
ステップ4
パスワード管理表からユーザ名useridに
相当する秘密情報passwordを取り出す
useridに対応する
秘密情報あり？
No
Yes
ステップ5
useridcapabilityとpasswordをビット連結して
一方向性関数を適用し、結果をdigestとする
ステップ6
digestとcapabilityMDは一致？
No
Yes
ステップ7
要求は権限情報capability
で許可されている？
No
Yes
終 了
（正当アクセス）
終 了
（不正アクセス）

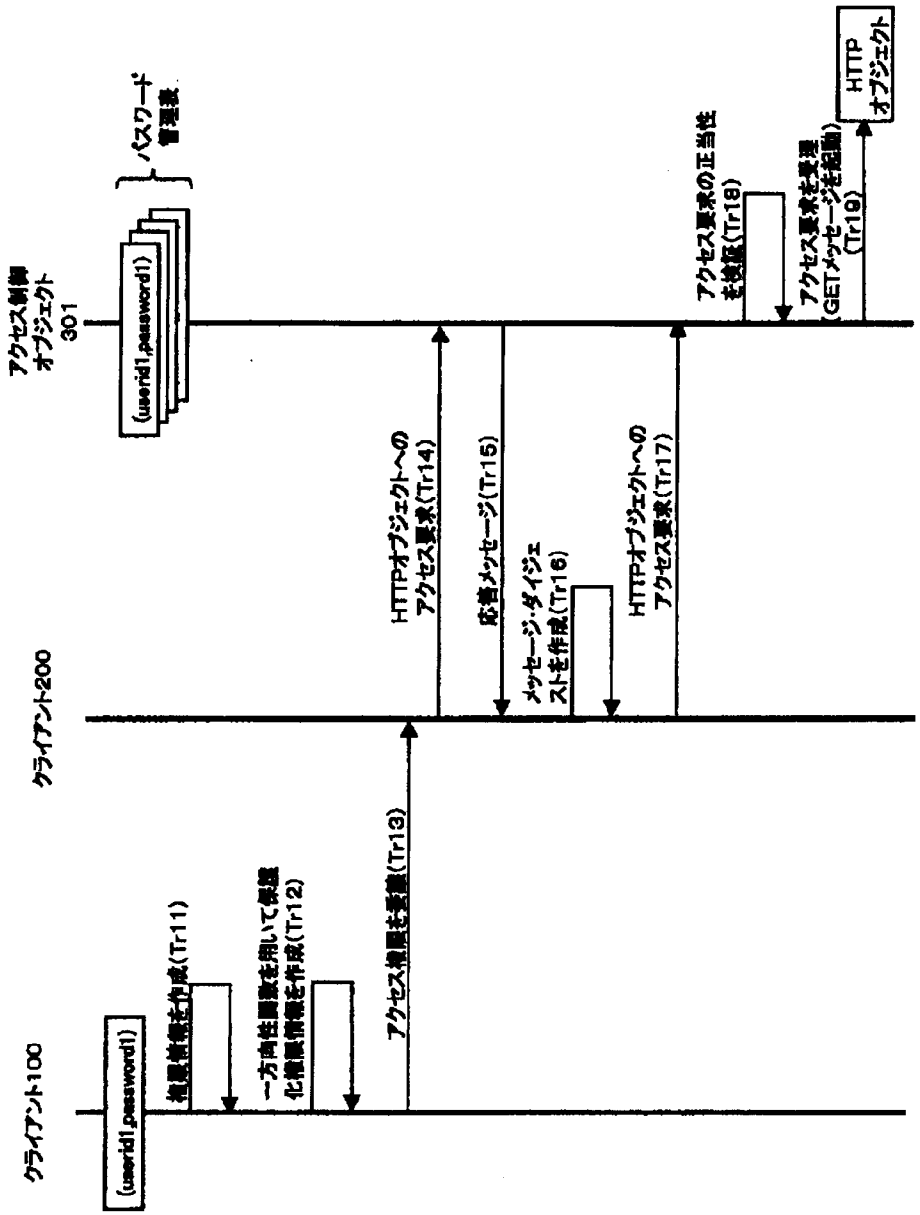
【図３】


クライアント100
(userid1,password1)
認証情報を作成(Tr1)
一方向性関数を用いて保護化認証情報を作成(Tr2)
アクセス情報を送信(Tr3)
クライアント200
HTTPオブジェクトへのアクセス要求(Tr4)
アクセス制御オブジェクト
301
(userid1,password1)
パスワード管理表
アクセス要求の正当性を検証(Tr5)
アクセス要求を受理(GETメッセージを起動)(Tr6)
HTTPオブジェクト

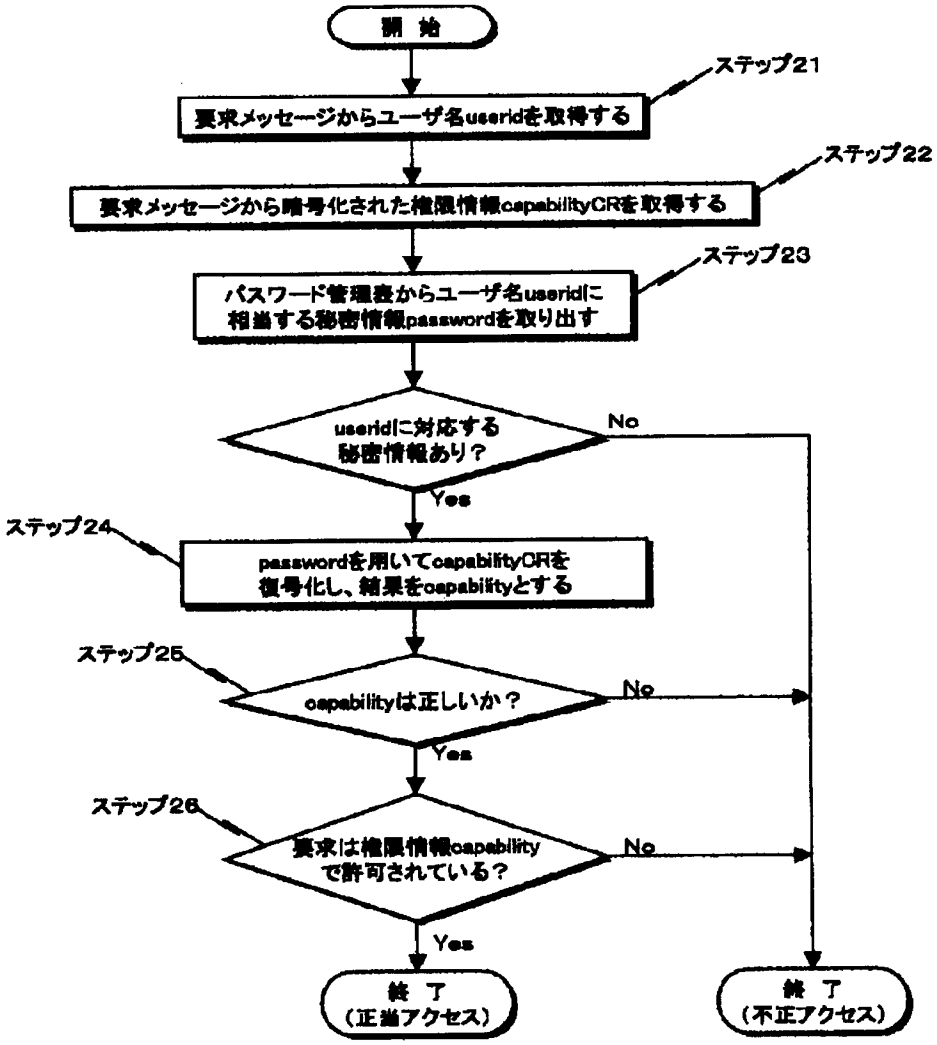
【図４】


開 始
ステップ11
要求メッセージからユーザ名useridを取得する
ステップ12
要求メッセージから権限情報capabilityを取得する
ステップ13
要求メッセージから権限情報のメッセージ・ダイジェストresponseを取得する
ステップ14
クライアント200に送信したチャレンジ文字列をnonce1とする
ステップ15
パスワード管理表からユーザ名useridに
相当する秘密情報paseswordを取り出す
useridに対応する
秘密情報あり？
No
Yes
ステップ16
チャレンジ文字列nonce1を用いて保護化
権限情報を再現し、結果をdigestとする
ステップ17
digestとresponseは一致？
No
Yes
ステップ18
要求は権限情報capability
で許可されている？
No
Yes
終 了
（正当アクセス）
終 了
（不正アクセス）

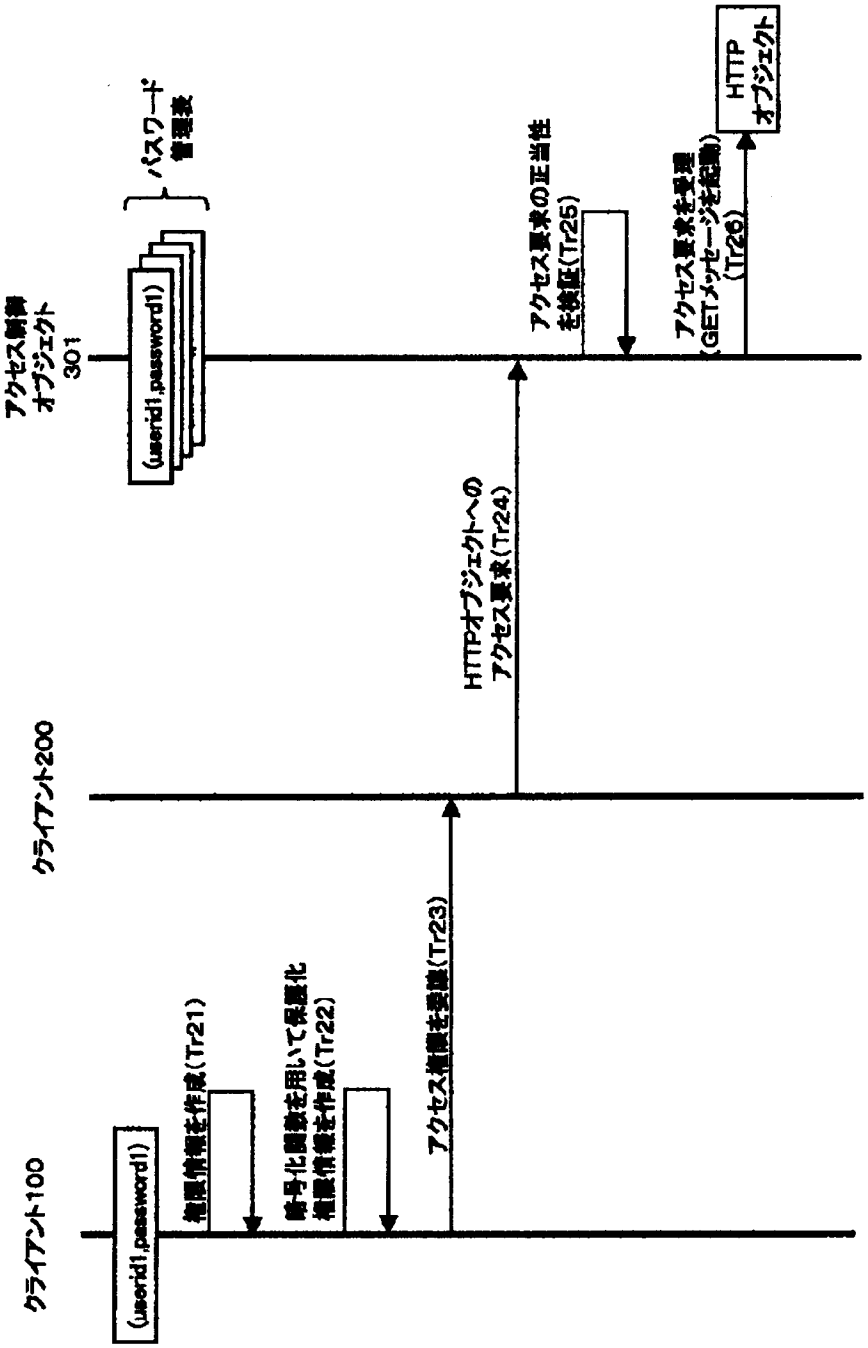
【図５】


クライアント100
クライアント200
アクセス制御
オブジェクト
301
パスワード
管理表
(userid1,password1)
(userid1,password1)
一方向性関数を用いて保護
化情報情報を作成(Tr12)
アクセス情報を送信(Tr13)
HTTPオブジェクトへの
アクセス要求(Tr14)
応答メッセージ(Tr15)
メッセージ・ダイジェ
ストを作成(Tr16)
HTTPオブジェクトへの
アクセス要求(Tr17)
アクセス要求の正当性
を検証(Tr18)
アクセス要求を受理
(GETメッセージを起動)
(Tr19)
HTTP
オブジェクト

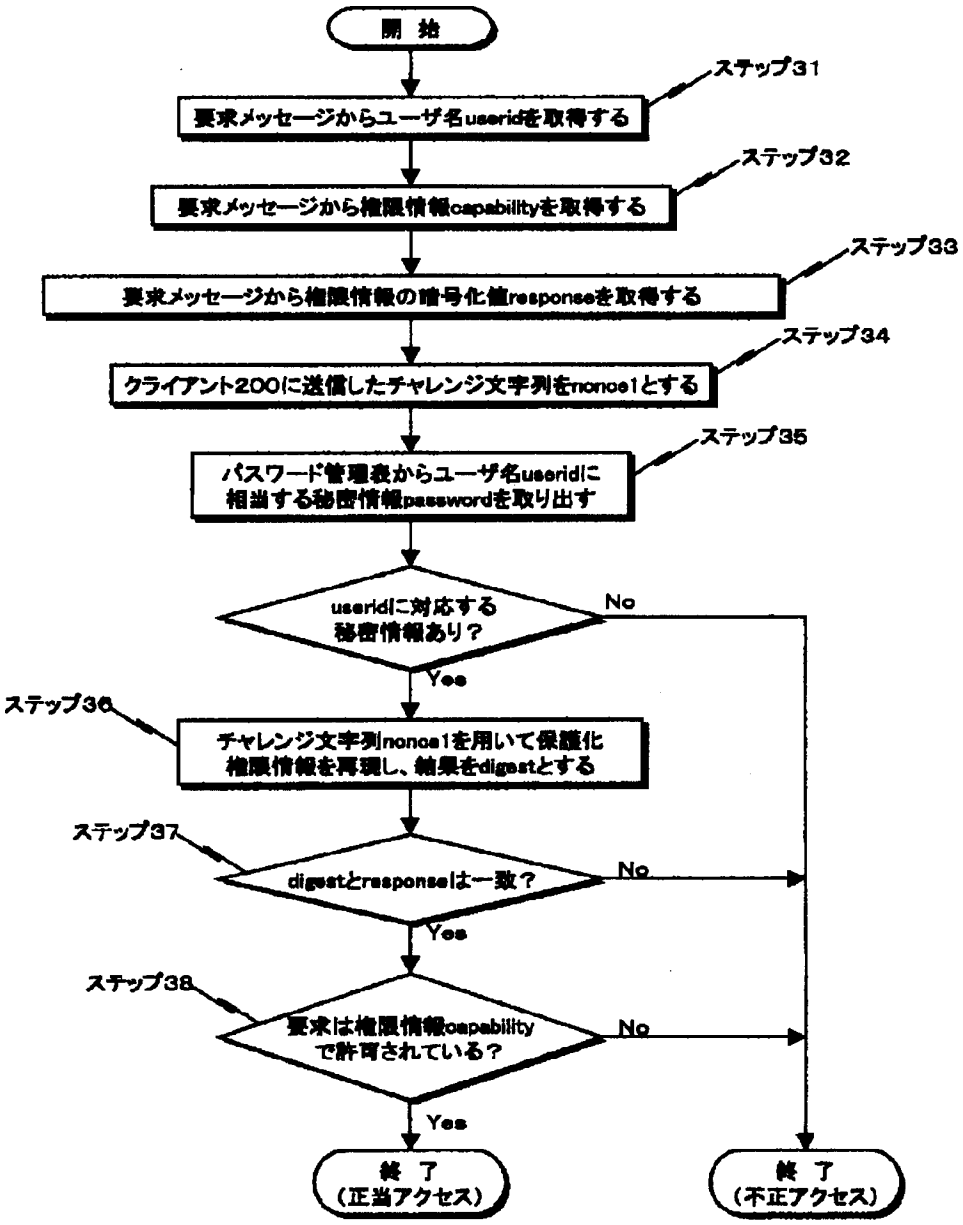
【図６】


開 始
ステップ21
要求メッセージからユーザ名useridを取得する
ステップ22
要求メッセージから暗号化された権限情報capabilityCRを取得する
ステップ23
パスワード管理表からユーザ名useridに
相当する秘密情報passwordを取り出す
useridに対応する
秘密情報あり？
No
Yes
ステップ24
passwordを用いてcapabilityCRを
復号化し、結果をcapabilityとする
ステップ25
capabilityは正しいか？
No
Yes
ステップ26
要求は権限情報capability
で許可されている？
No
Yes
終 了
（正当アクセス）
終 了
（不正アクセス）

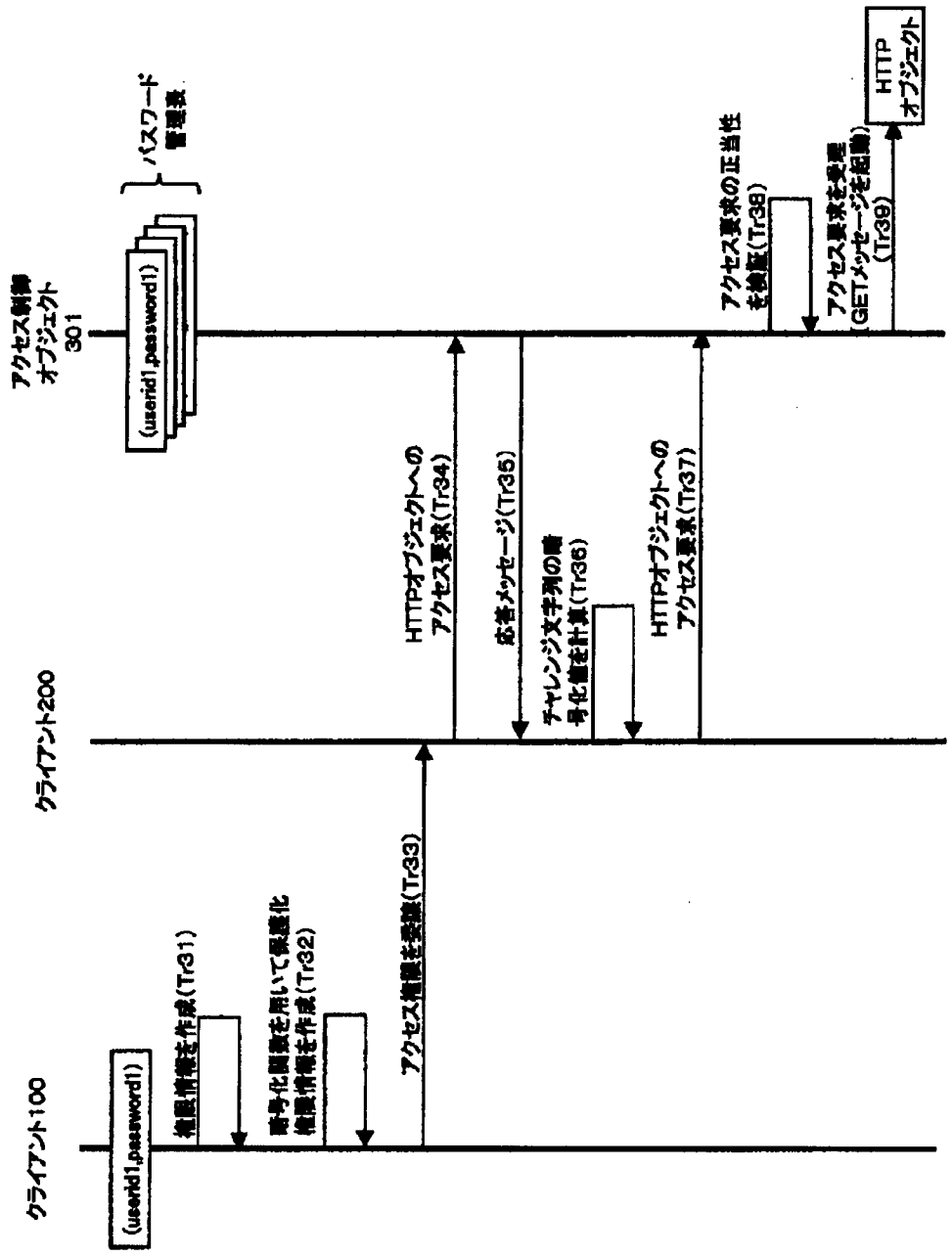
【図７】


クライアント100
クライアント200
アクセス制御
オブジェクト
301
パスワード
管理表
(userid1,password1)
(userid1,password1)
権限情報を作成(Tr21)
暗号化鍵を用いて保護化
権限情報を作成(Tr22)
アクセス権限を登録(Tr23)
HTTPオブジェクトへの
アクセス要求(Tr24)
アクセス要求の正当性
を検証(Tr25)
アクセス要求を受理
(GETメッセージを起動)
(Tr26)
HTTP
オブジェクト

【図８】


開始
ステップ31
要求メッセージからユーザ名useridを取得する
ステップ32
要求メッセージから権限情報capabilityを取得する
ステップ33
要求メッセージから権限情報の暗号化値responseを取得する
ステップ34
クライアント200に送信したチャレンジ文字列をnonce1とする
ステップ35
パスワード管理表からユーザ名useridに相当する秘密情報passwordを取り出す
useridに対応する秘密情報あり？
No
Yes
ステップ36
チャレンジ文字列nonce1を用いて保護化権限情報を再現し、結果をdigestとする
ステップ37
digestとresponseは一致？
No
Yes
ステップ38
要求は権限情報capabilityで許可されている？
No
Yes
終了（正当アクセス）
終了（不正アクセス）

【図９】


クライアント100
(userid1,password1)
認証情報を作成(Tr31)
暗号化関数を用いて認証化
認証情報を作成(Tr32)
アクセス権限を要求(Tr33)
クライアント200
HTTPオブジェクトへの
アクセス要求(Tr34)
応答メッセージ(Tr35)
チャレンジ文字列の暗
号化値を計算(Tr36)
HTTPオブジェクトへの
アクセス要求(Tr37)
アクセス制御
オブジェクト
301
(userid1,password1)
パスワード
管理表
アクセス要求の正当性
を検証(Tr38)
アクセス要求を承認
(GETメッセージを起動)
(Tr39)
HTTP
オブジェクト